MOBILE ROBOT NAVIGATION FOR BUILDING QUALITY ASSESSMENT

NIK MUHAMMAD ASYRAF BIN ABDULLAH SUHAIMI

BACHELOR OF MECHATRONIC ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

MOBILE ROBOT NAVIGATION FOR BUILDING QUALITY ASSESSMENT

NIK MUHAMMAD ASYRAF BIN ABDULLAH SUHAIMI

A report submitted
in partial fulfilment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Technology Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I declare that this thesis entitled "MOBILE ROBOT NAVIGATION FOR BUILDING QUALITY ASSESSMENT is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

Signature	:	
Name	:	NIK MUHAMMAD ASYRAF BIN ABDULLAH SUHAIMI
Date	:	25/1/2024

APPROVAL

I hereby declare that I have checked this report entitled "MOBILE ROBOT NAVIGATION FOR BUILDING QUALITY ASSESSMENT", and in my opinion, this thesis fulfils the partial requirement to be awarded the degree of Bachelor of Mechatronics Engineering with Honours

Signature :

Supervisor Name : PROF. MADYA. IR. DR. AHMAD ZAKI BIN HAJI SHUKOR

Date : 25/1/2024

DEDICATIONS

To my beloved mother and father

To all my respected teachers and friends

ACKNOWLEDGEMENTS

In preparing this report, I was contact with many people, researchers, academicians and practitioners. They have contributed towards my understanding and thought. First and foremost, I would like to take this opportunity to express my greatest gratitude to the Almighty for the blessing in completing my final year project successfully without any problem. Without good health from the Almighty I think this study not success until not. This success and final outcome of this project need a lot of guides from multiple individuals and I also want to give them thank for helping for complete this study.

I would like to express my appreciation to my supervisor, Prof. Madya. Ir. Dr. Ahmad Zaki bin Haji Shukor for his commitment, guidance, tips, and idea toward me during complete my study in final year project. I also want say thank you to my panels for Final Year Project Seminar, Prof. Madya. Dr. Ahmad Anas bin Yusof and Dr. Hairol Nizam bin Mohd Shah for the guide, critics and advices to improve my project. I also owe my deep gratitude to final year project coordinator, person that guide along by provide all necessary information to develop a good thesis.

Thank you also to all my friends, those who give help, suggestion, advice and motivation during completing this study. Without yours support I think I will give up. Special thanks to my beloved family for their non-stop support. They motivate myself to consistence do this study. Last but not least, thank you to University Teknikal Malaysia Melaka (UTeM) to give me change to use the facility and specialist to finish my final year project.

ABSTRACT

Building Quality Assessment is needed to ensure that every building construction is rated in terms of quality. For Building Quality Assessment, there are two parts which are architectural works and external works. For internal works, some of the criteria assessed are ceiling, wall, and floor. For wall quality assessment, human assessors normally use tools to assist in the assessment. Other than that, visual assessment is also done. However, human assessors take a long time to finish assessment for a large number of sample houses. Visual assessment, for example assessing the finishing of wall will be affected by the sunlight and tiredness of assessor. It is desired that an automatic wall data collection autonomous robot could help to navigate in a room to take data. The objectives of this project are to design a mobile robot that can move in a room with tiles and measure distance between the robot and wall, to develop algorithms for robot to follow the wall, and to evaluate the performance of robot following the wall. This project is done by designing the robot in Fusion 360, to ensure the suitable size used, to fit in all sensors and actuator. Then, the actuators (motors), body material are selected for this robot. The schematic diagram was also completed using the Fritzing software after assembling the robot hardware. The robot uses Mecanum wheels to allow it to slide left and right. The sensors used are the ultrasonic sensor and Time-of-Flight sensor to detect and measure the distance between the sensor and the wall. This robot needs to move in two types of rooms, namely Rectangle Shape Room and L-Shape Room. In addition, the theoretical calculation of the robot movement was done considering room dimensions before collecting experimental data. After completing the experiments, it is found that the experimental data found and the calculated motion time varies. In the experiments, the robot movies in two delay sliding times, namely 400 ms and 800 ms to assess the completion time of the motion. The set delay time 400 ms to 800 ms, is used to complete slide lefts in one cycle of the room shape. The experiment was repeated for 10 times for every room shape. The results of the experiments show that by increasing the sliding motion delay, it significantly reduces the time required to complete the total of slide left motion for each wall and the time to complete one cycle in the rooms experimented.

ABSTRAK

Penilaian Kualiti Bangunan diperlukan untuk memastikan setiap pembinaan bangunan dinilai dari segi kualiti. Bagi Penilaian Kualiti Bangunan, terdapat dua bahagian iaitu kerja seni bina dan kerja luaran. Untuk kerja dalaman, beberapa kriteria yang dinilai adalah siling, dinding, dan lantai. Untuk penilaian kualiti dinding, penilai manusia biasanya menggunakan alat untuk membantu dalam penilaian. Selain itu, penilaian visual juga dilakukan. Walau bagaimanapun, penilai manusia mengambil masa yang lama untuk menyelesaikan penilaian bagi sejumlah besar rumah sampel. Penilaian visual, contohnya menilai kemasan dinding akan dipengaruhi oleh cahaya matahari dan keletihan penilai. Adalah diingini bahawa robot autonomi pengumpulan data dinding automatik boleh membantu menavigasi dalam bilik untuk mengambil data. Objektif projek ini adalah untuk mereka bentuk robot mudah alih yang boleh bergerak di dalam bilik dengan jubin dan mengukur jarak antara robot dan dinding, untuk membangunkan algoritma untuk robot mengikuti dinding, dan untuk menilai prestasi robot mengikuti dinding. Projek ini dilakukan dengan mereka bentuk robot dalam Fusion 360, untuk memastikan saiz yang sesuai digunakan, untuk dimuatkan dalam semua sensor dan penggerak. Kemudian, penggerak (motor), bahan badan dipilih untuk robot ini. Gambarajah skematik juga telah dilengkapkan menggunakan perisian Fritzing selepas memasang perkakasan robot. Robot menggunakan roda Mecanum untuk membolehkannya meluncur ke kiri dan kanan. Penderia yang digunakan ialah penderia ultrasonik dan penderia Time-of-Flight untuk mengesan dan mengukur jarak antara penderia dan dinding. Robot ini perlu bergerak dalam dua jenis bilik iaitu Rectangle Shape Room dan L-Shape Room. Selain itu, pengiraan teori pergerakan robot dilakukan dengan mengambil kira dimensi bilik sebelum mengumpul data eksperimen. Setelah selesai menjalankan eksperimen, didapati data eksperimen yang ditemui dan masa gerakan yang dikira berbeza-beza. Dalam eksperimen tersebut, robot merakam dalam dua masa gelongsor kelewatan, iaitu 400 ms dan 800 ms untuk menilai masa siap gerakan. Masa tunda yang ditetapkan 400 ms hingga 800 ms, digunakan untuk melengkapkan kiri slaid dalam satu kitaran bentuk bilik. Eksperimen diulang sebanyak 10 kali untuk setiap bentuk bilik. Keputusan eksperimen menunjukkan bahawa dengan meningkatkan kelewatan gerakan gelongsor, ia mengurangkan dengan ketara masa yang diperlukan untuk melengkapkan jumlah gerakan kiri gelongsor bagi setiap dinding dan masa untuk melengkapkan satu kitaran di bilik yang dieksperimen.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Wi-Fi	-	Wireless Fidelity
UART	-	Universal Asynchronous Receiver-Transmitter
USB	-	Universal Serial Bus
HDMI	-	High-Definition Multimedia Interface
DIY	-	Do-it-yourself
V	-	Voltage
GPS	-	Global Positioning System
Lidar	-	Light Detection and Ranging
SLAM	-	Simultaneous Localization and Mapping
IR	-	Infrared
BLE	-	Battery Low Energy
LRF	-	Laser Range Finder
LED	-	Light Emitting Diode
RGB	-	Red, Green Blue
IOT	-	Internet Of Thing
ToF	-	Time of Flight

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Innovative construction building equipment that called mobile robot navigation for building quality assessment to enable the robot to make inspection in construction environment for building. The robot has sensors, actuator, and motors that give them the ability to navigate autonomously within a building environment, capturing relevant data and performing quality checks on various aspects. Mobile robot navigation can make more efficient, accurate, and cost effective for building assessment process.

Traditionally, building quality assessment has relied on manual inspections, which can be time-consuming, costly, and prone to human error. However, this method requires a lot of work and is sometimes inaccurate, resulting in the construction of buildings process. Mobile robot navigation provides a more accurate and potential to automate the assessment process, reduce human intervention, and provide accurate and objective measurements.

Additionally, mobile robot navigation can increase precision and efficiency in construction building inspection process. Mobile robot navigation can also improve the safety of construction workers by reducing their exposure to hazardous environments. Last but not least, the accuracy and effectiveness of mobile robot navigation processes can have a positive impact on building construction by reducing the risk of human carelessness recording data and improve the building's quality standards.

In conclusion, the development of a mobile robot navigation system for building quality assessment holds great potential to revolutionize the way we evaluate and ensure the quality of buildings. By leveraging advanced sensors, navigation algorithms, and autonomous capabilities, this project aims to enhance the efficiency, accuracy, and cost-effectiveness of the assessment process, ultimately leading to safer, more functional, and higher-quality buildings.

1.2 Problem Statement

The construction industry struggles with a number of obstacles and constraints that limit the effective and timely completion of building projects. One key issue is the lack of reliable and automated ways for assessing construction quality during the construction process. Manual inspections are time-consuming, labour-intensive, and prone to human error in traditional evaluation procedures. These constraints cause delays, cost overruns, and potential safety dangers because errors and issues may go undetected until later phases or even after the project is completed. To solve these problems, there is a need to design a Mobile Robot Navigation For Building Quality Assessment that can automate and expedite the process of inspecting and assessing construction quality in real-time, ensuring early detection of issues, greater efficiency, and improved overall building quality.

1.3 Project Objectives

To create a mobile robot navigation that will help construction industry, these following objectives are proposed:

a) To design a mobile robot that can move in a room with tiles and measure distance by the way robot and wall.
b) To develop algorithms for robot to follow the wall.
c) To evaluate the performance of robot following the wall.

1.4 Project Scope

The scope of the project will focus on development motion mechanisms of wheels and sensor for the robot. The scopes and limitations are:

- The motion mechanism of wheels the robot mainly focusses on environment of building area and room size.
- The movement mobile robot needs appropriate software.
- The robot must use a good microcontroller which have many inputs and output as Arduino.
- The accuracy of sensor for the robot to perform the movement robot more efficiency.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will discuss the previous study related or having something that similar to the project. The aim or function of the previous study that been focused on the development of motion mechanism of wheels and component of the system for the mobile robot. From all the information from the previous study, there are some information that need to be highlight and discuss in the last part of this chapter for this project to be success. This previous study will help in expending ideas of the project and broaden the knowledge about the things that related to the project.

2.2 The Motion Mechanism of Wheels

The motion mechanism of the wheels is a critical component in the navigation of a mobile robot, as it determines the direction and speed of the robot's movements. It’s also had a complex system that involves various sensors, algorithms, and actuators working together to enable the robot to move through its environment with precision and accuracy. The algorithms that are related like a fuzzy logic-based local optimisation navigation method that processes sensor data using a "recognition-memory" approach. Simulation studies demonstrate that the mobile robot, when guided by fuzzy rules, can avoid the dead zone [1]. The combination of mechanical design on the wheel and chassis, motion control and multiple input/output sensors allow the exploration of large number of control algorithm and software to be implemented to the robot for practical applications [2]. Can be said that three wheels are better than four wheels for mobile robots [3]. But the other ways use of standard or fixed wheels in four-wheel steering system has better efficiency and accuracy [4]. For the other conditions to control movement wheels on mobile robot navigation, wireless controller with mobile base was suggested for the mobile robot. The current generation of high-end mobile phones are ready to execute complex and demanding algorithms for mobile robot control [5]. By making control robot using virtual paths on the Android smartphone which have an accuracy of more than 70% to a scale of 1:50 with the actual track in an environment that is not wide enough, the advantage can be used to control the robot move easily and efficiently compared to the usual manual control robot that will take more time to control the robot moves its place as desired. [6]. There are several types of connectivity between robot with smartphone that can be applied.

2.2.1 Connectivity using Wi-Fi Module

The connectivity using a Wi-Fi module, allowing users to remotely control and monitor their devices from a smartphone or computer. The modern service business is evolving in large part thanks to the internet of things, a new technology system made up of several information technologies. Using this method, mobile robot can be controlled via the web, from any places connected to the network without constructing specific infrastructures for communication [7]. This mobile robot can be operated from everywhere in the world by using internet of things (IOT) [8]. The Wi-Fi Module on robot that has been designed is capable of remote operation from any location within the wireless coverage area, if there is an Android device running the control system software. This enables users to control and monitor the robot from a distance, providing a greater level of convenience and flexibility in its operation. The system for remote monitoring and control of industrial robots, which utilizes an Android device and Wi-Fi communication. The system enables users to intuitively control the robot from a great distance while simultaneously monitoring its motion by observing a 3D model of the robot's movement or trajectory [9].

2.2.2 Connecting using Bluetooth Module

Mobile Robot also can be reached using Bluetooth module connectivity to secure the connection between smartphone and robot. This control method can be implemented utilizing an android-controlled mobile phone that is mirrored via an internet server to control the robot using Bluetooth and receive input through the application [10]. Robot motion can be controlled in accordance with commands received from the android by using a Bluetooth module to connect the microcontroller and android via the Universal Asynchronous Receiver-Transmitter (UART) protocol [11]. Using asynchronous serial communication at a speed that can be adjusted, UART is a hardware communication mechanism. Asynchronous refers to the absence of a clock signal that would have allowed the output bits from the transmitting device to be synchronized with those at the receiving end [12]. Otherwise, mobile robot can be communication between the cell phone and the Arduino BT board through wireless [13]. The Bluetooth connection between the Smartphone and the Bluetooth robot can be fully controlled by a range of 25 meters, for a distance of 25-32 meters has a signal drop and is broken, and more than a range of 32 meters will experience a broken connection so that the robot cannot be controlled anymore [14]. The usage of Bluetooth enables accurate control over the robot's motions by providing a dependable and effective method of communication between the robot and the controller.

2.3 Component of The System

A system is made up of a number of parts that cooperate to carry out the project's objective. Hardware such as sensors and actuators as well as software such as operating systems and algorithms can be included in these parts. The precise function and goal of a system will determine its exact components. The mobile robot may contain sensors for data collection, a microcontroller for data processing, and a cloud platform for data storage and analysis. Designing and troubleshooting a system require an understanding of its components. There are several components that must be used to design the system for ensuring that robot can be function with maximum effectiveness in the agriculture field.

2.3.1 Raspberry Pi

A The Raspberry Pi is a well-liked option for use in robotics projects because of its compact size, low price, and versatility. It can serve as a robot's primary control board, providing the connectivity and computing power required to run all of the robot's systems and parts. The Raspberry Pi's variety of inputs and outputs, including Universal Serial Bus (USB) ports and High-Definition Multimedia Interface (HDMI), make it simple to connect to sensors, motors, and other peripherals, enabling the robot to collect information from its environment and interact with its surroundings [15]. One can develop a wide range of complex and intelligent systems that are capable of carrying out a number of activities and functions by employing a Raspberry Pi as the central controller for a robot.

2.3.2 Arduino UNO / MEGA

A popular open-source microcontroller platform for IoT and DIY electronics applications is Arduino. It is a well-liked option for both inexperienced and seasoned makers due to its user-friendly hardware and software development environment and extensive user and developer community. An open-source microcontroller development board is called an

Arduino. The Arduino may be used to read sensors and control devices like motors and lights, to put it simply. This enables programmes to be submitted to this forum and interact with real-world objects. The robot will reply and respond to the world at large using this [16]. IoT projects can be built with Arduino boards since they include a few form factors and connectivity options, such as Wi-Fi and Bluetooth.

2.3.3 Sensor

A sensor is an electronic device or module that detects and measures physical quantities or environmental conditions. It converts the physical input into an electrical signal or data that other systems or devices can process and use. Sensors are utilised in a variety of sectors and applications, including as robotics, industrial automation, healthcare, environmental monitoring, and consumer electronics. Temperature sensors, pressure sensors, motion sensors, light sensors, proximity sensors, and many other types of sensors are examples. They are critical in gathering real-world data and allowing interaction between the physical and digital worlds. Sensors are categorised into many types based on their uses, input signal, and conversion mechanism, as well as features such as cost, accuracy, and range [17].

2.3.3.1 Ultrasonic Sensor

Ultrasonic sensors work by transmitting a pulse of sound, much like sonar detectors, outside the range of human hearing. This pulse travels away from the range finder in a conical shape at the speed of sound (340 m/s). The sound reflects off an object and back to the range finder. The sensor interprets this as an echo and calculates the time interval between sending the signal and receiving the echo. This interval is then computed by a controller to determine the distance of the object [18]. The use of ultrasonic sensors is extremely popular among sensor products. The use of this sensor is more to detect the distance based on a specific area and most of these sensors are used on robots because the robots need some sense to prevent them from colliding.

2.3.3.2 RPLidar AL Sensor Gyroscope

RPlidar A1 adopts laser triangulation technology, and with SLAMTEC's high-speed vision acquisition and processing mechanism, it can measure the distance data of more than

8000 times/second. The RPlidar A1 utilizes a ranging that rotates clockwise, enabling 360° full-scan detection of the surrounding environment and producing a map of the area and ideal for Robot Navigation and Localization [19]. Users can observe the cloud map of the environment scanning point collected by the RPLIDAR A1 [20]. Mobile robot can perform this sensor for testing machine learning accuracy using SLAM [21].

2.3.3.3 Infrared Sensor

An infrared sensor (IR sensor) is a radiation-sensitive optoelectronic component with a spectral sensitivity in the infrared wavelength range 780 nm until 50 µm. IR sensors are now widely used in motion detectors, which are used in building services to switch on lamps or in alarm systems to detect unwelcome guests [22].

2.3.3.4 MPU9250 IMU Sensor

The MPU9250 is a 9-axis inertial measurement unit (IMU) sensor that includes an accelerometer, gyroscope, and magnetometer. It is commonly used in embedded systems and sports monitoring to determine body orientation and movement. The sensor's data can be transferred wirelessly using Bluetooth Low Energy (BLE) technology [26]. The MPU9250 can be used in gesture detection systems, where the attitude angle is measured using the internal sensors and data fusion analysis is performed using the Kalman filter algorithm [27].

2.3.3.5 Hokuyo URG-04LX-UG01 Laser Range Finder (LRF)

The papers provide a detailed characterization of the Hokuyo URG-04LX-UG01 laser range finder (LRF) and its applications [28] and [29] compare the Hokuyo LRF with the Sick LRF, which has been the standard range sensor for mobile robot obstacle avoidance and mapping applications. They found that the Hokuyo LRF is smaller, lighter, and consumes less power, making it more suitable for small mobile robots. However, the accuracy of the Hokuyo LRF is strongly dependent on the target properties, such as color, brightness, and material, and the incidence angle proposed a calibration model to measure the accurate distance to a target [30].

2.3.3.6 Inductive Proximity Sensor

An inductive proximity sensor [31] describes an integrated inductive proximity sensor with high sensitivity and submicrometric resolution, and also discusses its temperature behaviour and compensation scheme. That is proposes the use of a magneto plated wire as a sensing coil to extend the operating distance of inductive proximity sensors [32].

2.4 Table Comparison of Sensors

Table 2.1: Comparison of sensors

Sensor	**Specifications**	**Range Distance**
Ultrasonic Sensor [33]	Emits ultrasonic waves, measures bounce-back time	2 cm to 4 meters
RPLidar A1 Sensor Gyroscope [34]	Laser scanner, uses gyroscope for rotation	Up to 25 meters
Infrared Sensor [35]	Detects infrared radiation, uses LED and photodiode	5 cm to 20 cm
Kinect Sensor [36]	Depth-sensing, combines RGB and structured light	0.5 meters to 4 meters
Hokuyo URG-04LX-UG01 laser range finder (LRF) [37]	Laser range finder with 240-degree scanning	Up to 4 meters
Inductive Proximity Sensor [38]	Detects metallic objects using electromagnetic fields	2 mm to 10 cm

The table above contains information about the various sensors. An ultrasonic sensor emits ultrasonic waves and measures the time it takes for the waves to recover. It offers a detection range of 2 cm to 4 meters. The RPLidar A1 Sensor Gyroscope is a laser scanner that uses a gyroscope for rotation and offers a detection distance of up to 25 meters. The infrared sensor detects infrared radiation with an LED and a light-emitting diode in the range of 5 cm to 20 cm. The Kinect Sensor can detect depth by combining RGB and structured light and offers a range from 0.5 meters to 4 meters. The Hokuyo URG-04LX-UG01 Laser Range Finder (LRF) scans 240 degrees and can detect targets up to 4 meters away. The inductive proximity sensor detects metal objects using electromagnetic fields and has a range of 2 mm to 10 cm. Among the sensors listed in the table, Ultrasonic sensors stand out as the best and most cost-effective choice. The ability to measure ultrasonic transmission and recovery time provides a versatile and reliable solution for distance measurement. Ultrasonic sensors not only offer a wide detection range from 2 cm to 4 meters and cover short or long distances but are also known for their affordability.

Compared to other sensors, such as the RPLidar A1 sensor gyroscope, which excels in remote sensing but is expensive, or the Kinect sensor, which specializes in depth sensing but is relatively expensive, ultrasonic sensors offer a good balance of performance and performance. to enlarge performance and affordability. Its simplicity, reliability and wide availability make it popular in various applications such as robotics, object detection and proximity detection, making it the best and most cost-effective choice among the listed sensors.

2.5 Summary of Literature Review

A thorough summary of all previous research on a subject is called a literature review. It is intended to pinpoint gaps and potential topics for additional research and tries to provide a critical analysis of the available data and expertise on a subject. This summary will select the finest alternative to successfully complete the mission of the mobile robot navigation. There are few main previous research that been focused to:

Table 2.2: Summaries of 15 main previous research.

Criteria / Journals	Type of Sensor	Type of Microcontroller	Type of software	Purpose	Measure Parameters Performance Index	Motion Mechanism	Results
1. Developing Design of Mobile Robot Navigation System Based on Android [39]	Smartphone gyro sensor	Arduino UNO	Remotexy.com (HC05 Bluetooth communication system)	Phone control mobile robot navigate autonomously in indoor environments	Observation of robot motion. For distance of Bluetooth from robot	Two wheels DC motor	The range of Bluetooth-based control reaches a maximum distance of 10.5 meters
2. A Semantic Classification Approach for Indoor Robot Navigation [40]	RPLidar AI sensor gyroscope	Raspberry Pi	Python	Testing machine learning accuracy using SLAM	Measure accuracy of machine learning model	Four wheels track	The system achieved 95.8% accuracy in classifying objects as navigable or non-navigable

3. Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation [41]	Ultrasonic sensor Infrared sensor	Raspberry Pi 3 and Arduino UNO ATmega328P	Arduino IDE and Python	Line following in indoor environment	The accuracy of the system's navigation, obstacle avoidance, and decision-making capabilities	Four wheels DC motor	Ability to navigate through a test environment and evade obstacles
4.Mobile robot navigation in unknown static environments using ANFIS controller [42]	One Ultrasonic and two Infrared	Arduino ATmega328P	MATLAB and C/C++	Obstacles avoidance mobile robot	The ANFIS controller is effective in guiding the robot in the given environments without any collision	Two wheels	Successful simulation and experimental results on actual mobile robot
5. Obstacle detection using ultrasonic sensor for a mobile robot [43]	Ultrasonic HC-SR04	Arduino UNO R3	Arduino IDE	Testing ultrasonic sensor on different obstacles type	Ability to accurately detect and avoid obstacles in various environments	-	Positions of obstacles showed the flexibility of the robot to avoid
6. Wheeled Mobile Robot Obstacle Avoidance Using Compass and Ultrasonic [44]	Ultrasonic, Infrared and Compass	Arduino Mega 2560	C language (Arduino IDE)	Obstacles avoidance mobile robot with compass	The robot was able to navigate autonomously to the north direction and avoid obstacles in an outdoor environment	Two wheels DC motor	The robot was able to move freely in an unknown environment and avoid obstacles in its path

7. Design of mobile robot navigation controller using neuro-fuzzy logic system [45]	Kinect sensor	-	The virtual robot experimentation platform (V-REP), ROS Groovy Galapagos, and SolidWorks	Purposeful and reactive navigation while complying with safety laws in organized environments	Ability to navigate the robot safely and efficiently in simulated environments	Four wheels	Testing and evaluating the performance of the designed controller in terms of the robot's ability
8. Experimental study of terrain coverage of an autonomous chaotic mobile robot [46]	Ultrasonic sensor HC-SR04 and two Infrared	Arduino UNO	Arduino IDE	To locate and avoid the boundaries of the workspace	Appropriate sensor for avoiding collision with the boundaries	Two wheels motor with gearbox	Fast scanning of the robot's workspace with unpredictable way

9. Metric-Cauchy sequence for wheeled mobile robot navigation [47]	In-house robot: a) Infrared sensors with up to 150 cm range. b) Ultrasonic sensors with up to 400 cm range Khepera-II robot: 8 Infrared proximity and ambient light sensors with up to 100 mm range	In-house robot: ATmega2560 (Arduino Mega 2560, Arduino UNO) Khepera-II robot: Motorola 68,331 CPU, 25 MHz	C/C++ and MATLAB	To present a path planning approach for wheeled mobile robot navigation using Metric-Cauchy Sequence for static environment	The percentage of deviation between simulation and experimental results of path length and navigational time	In-house robot: 2-DC gear motors with incremental encoders Khepera-II robot: 2-DC brushed Servo motors with incremental encoders	Controller has demonstrated successful navigation in both simulation and experimental

10. Collision avoidance approaches for autonomous mobile robots to tackle the problem of pedestrians roaming on campus road [48]	LM393 Hall Effect Sensors, MPU9250 IMU sensors	Jetson Xavier NX, Arduino UNO and Intel Realsense D435	Arduino IDE and MATLAB/Simulink	Able to avoid collisions while navigating the mobile robot	To provide a solution to the problem of collision avoidance for autonomous mobile robots in real-world scenarios with pedestrians	Four wheels	Proposed collision avoidance algorithm for autonomous mobile robots
11. Performance Evaluation of Ultrasonic and Infrared Waves on Human Body and Metal Surfaces for Mobile Robot Navigation [49]	Three Ultrasonic HC SR-04 and three Infrared SHARP GP2D12	Arduino MEGA	MATLAB and Simulink	To improve the effectiveness of robotic navigation and obstacle avoidance	Correlation values, Root mean square error (RMSE) and Mean absolute percentage error (MAPE)	Two wheels	Ultrasonic sensors performed better than infrared sensors

12. An Inexpensive Autonomous Mobile Robot for Undergraduate Education: Integration of Arduino and Hokuyo Laser Range Finders [50]	Hokuyo URG-04LX-UG01 laser range finder (LRF)	Arduino an ESP-WROOM-02 microcontroller from Espressif Systems, Shanghai, China	Arduino IDE	To present a framework for teaching mechatronics to undergraduate mechanical engineering students using an inexpensive autonomous mobile robot	The students' improvement in technical knowledge, motivation, interest, and satisfaction after completing the course effectiveness	Omnirover 3WD	Given to the student participants upon completion of the course indicated.
13. Vision-Based Mobile Robot Navigation For Suspicious Object Monitoring In Unknown Environments [51]	Gas Sensor: MQ-136 Hydrogen and Sulfide Metal Sensor: Inductive Proximity	Arduino Mega 2560	Visual studio 2015 and Arduino IDE (XBee and USB Wi-Fi TL-WN8200ND for serial communication)	To detect suspicious objects in unknown environments	Measure accuracy distance for gas sensor and metal sensor	Four wheels DC motor track	Developed mobile robot navigation system can be an effective tool

14. Embedded Design and Implementation of Mobile Robot for Surveillance Applications [52]	Ultrasonic sensor and Temperature sensor (DS18B20)	Arduino and Raspberry Pi (Arduino microcontroller is used to retrieve the robot movement command sent by the operator and translate it into signals used to drive robot motors and Raspberry Pi is used for software development for controlling the robot	Arduino IDE	To improve the surveillance and security of areas such as homes, laboratories, offices, factories, and airports to prevent any threatening to human lives as well as collecting and transmitting data in real-time	The speed of the robot, the accuracy of the obstacle detection system, and the reliability of the wireless communication system	Four wheels DC motor	The robot is able to perform various tasks
15. An Android Based Mobile Robot for Monitoring and Surveillance [53]	Ultrasonic sensor	Arduino UNO	C++ and Arduino IDE	Surveillance and monitoring	Accuracy (number of successful avoidances/numbers of test cases)	Four-wheeled DC motors	The robot demonstrated flexibility in avoiding obstacles and had a communication range of nearly 50m.

Journal 1, "Developing Design of Mobile Robot Navigation System Based on Android," utilizes a smartphone gyro sensor for control, allowing autonomous navigation in indoor environments through Bluetooth communication. The advantage lies in the simplicity and accessibility of using a smartphone as a control interface. The disadvantage, the limited range of Bluetooth communication, restricting the robot's navigational capabilities.

Journal 2, "A Semantic Classification Approach for Indoor Robot Navigation," employs an RPLidar AI sensor and Raspberry Pi for testing machine learning accuracy using SLAM. The advantage here is the utilization of advanced machine learning techniques for accurate object classification. The disadvantage, the reliance on complex algorithms and hardware may make it more challenging to implement and maintain.

Journal 3, "Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation," combines ultrasonic and infrared sensors with Raspberry Pi and Arduino Uno for line following in indoor environments. This setup offers versatility in sensor options and the power of Raspberry Pi for processing. The disadvantage, the performance may be limited by the computational capabilities of the microcontrollers.

Journal 4, "Mobile Robot Navigation in Unknown Static Environments Using ANFIS Controller," utilizes an ANFIS controller with ultrasonic and infrared sensors for obstacle avoidance. The advantage of this approach is the effective guidance of the robot without collisions in unknown environments. The disadvantage, the reliance on ANFIS controllers and complex algorithms may require additional computational resources and expertise.

Journal 5, "Obstacle Detection Using Ultrasonic Sensor for a Mobile Robot," focuses on testing the ultrasonic sensor's ability to detect and avoid obstacles in various environments. The advantage lies in the simplicity and affordability of the ultrasonic sensor. The disadvantage, its range and accuracy may be limited compared to more advanced sensor options.

Journal 6, "Wheeled Mobile Robot Obstacle Avoidance Using Compass and Ultrasonic," combines ultrasonic, infrared, and compass sensors for obstacle avoidance in outdoor environments. The advantage here is the ability to navigate autonomously and avoid obstacles while considering directional information. The disadvantage, the integration of multiple sensors and the complexity of outdoor environments can pose challenges.

Journal 7, "Design of Mobile Robot Navigation Controller Using Neuro-Fuzzy Logic System," utilizes a Kinect sensor for purposeful and reactive navigation in organized environments. The advantage lies in the ability to navigate the robot safely and efficiently in simulated environments. The disadvantage, the lack of specific information regarding the type of microcontroller and software used limits further assessment.

Journal 8, "Experimental Study of Terrain Coverage of an Autonomous Chaotic Mobile Robot," employs an Arduino UNO with an ultrasonic sensor and two infrared sensors to locate and avoid boundaries in the workspace. The advantage here is the appropriate sensor selection for collision avoidance. The disadvantage, the lack of detailed results and comparison with other methods hinders a comprehensive evaluation.

Journal 9, "Metric-Cauchy Sequence for Wheeled Mobile Robot Navigation," presents a path planning approach using infrared and ultrasonic sensors for a static environment. The advantage lies in successful navigation demonstrated in both simulation and experimental scenarios. The disadvantage, specific performance metrics and comparisons are not provided.

Journal 10, "Collision Avoidance Approaches for Autonomous Mobile Robots to Tackle the Problem of Pedestrians Roaming on Campus Road," utilizes LM393 Hall Effect Sensors and MPU9250 IMU sensors with Jetson Xavier NX, Arduino UNO, and Intel Realsense D435. The advantage here is the proposed collision avoidance algorithm for real-world scenarios with pedestrians. The disadvantage, specific performance results and comparisons are not mentioned.

Journal 11, "Performance Evaluation of Ultrasonic and Infrared Waves on Human Body and Metal Surfaces for Mobile Robot Navigation," compares the effectiveness of ultrasonic and infrared sensors for obstacle avoidance. The advantage is the evaluation using correlation values, root mean square error (RMSE), and mean absolute percentage error

(MAPE). The finding that ultrasonic sensors perform better than infrared sensors provides valuable insights.

Journal 12, "An Inexpensive Autonomous Mobile Robot for Undergraduate Education: Integration of Arduino and Hokuyo Laser Range Finders," utilizes a Hokuyo URG-04LX-UG01 laser range finder with Arduino and ESP-WROOM-02 microcontroller. The advantage lies in presenting an inexpensive framework for teaching mechatronics to undergraduate students, leading to improved technical knowledge and motivation. The specific robot configuration and its performance are not elaborated upon.

Journal 13, "Vision-Based Mobile Robot Navigation for Suspicious Object Monitoring in Unknown Environments," employs gas and metal sensors with Arduino Mega 2560 for detecting suspicious objects. The advantage is the potential effectiveness of the developed mobile robot navigation system. The disadvantage, specific results and comparisons are not provided.

Journal 14, "Embedded Design and Implementation of Mobile Robot for Surveillance Applications," uses an ultrasonic sensor and a temperature sensor with Arduino and Raspberry Pi. The advantage here is the improvement of surveillance and security, along with real-time data collection and transmission. The disadvantage, specific performance results and comparisons are not mentioned.

Journal 15, "An Android-Based Mobile Robot for Monitoring and Surveillance," utilizes an ultrasonic sensor with Arduino UNO for surveillance and monitoring purposes. The advantage is the flexibility in avoiding obstacles and the extended communication range. The disadvantage, further details and comparisons are not provided.

The 15 journals provide a comprehensive overview of various sensor technologies used in mobile robot navigation. The sensors used include smartphone gyro sensors, RPLidar AI sensors, ultrasonic sensors, infrared sensors, compass sensors, gas sensors, temperature sensors, and laser range finders. These sensors serve different purposes such as obstacle detection and avoidance, boundary detection, path planning, surveillance, and object monitoring.

Among the sensor technologies, ultrasonic sensors emerged as a common choice, offering accurate obstacle detection and avoidance capabilities. They were found effective in various environments and performed better than infrared sensors in terms of navigation and obstacle avoidance. Additionally, the integration of multiple sensors, such as combining ultrasonic, infrared, and compass sensors, enhanced the overall navigation capabilities of the mobile robots. The use of sensor fusion techniques, including machine learning algorithms, was also explored to improve the accuracy and reliability of the sensor data for navigation.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The purpose of this methodology section is to ensure that the objectives are fulfilled by detailing out the required process or procedures for each objective clearly. The objectives mentioned in Chapter 1 are to design a mobile robot that can move in a room with tiles and measure distance by the way robot and wall, to develop algorithms for robot to follow the wall. This approach aims to evaluate the efficiency of robot following the wall. The methods to achieve Objective 1 will be explained in section 3.2 while objective 2 is explained in section 3.3 and objective 3 in section 3.4.

3.2 Design of a mobile robot capable of solving construction contractor problems

To achieve the objective of designing a mobile robot navigation for building quality assessment, there are five steps needed, and are explained in the subsections 3.2.1 until 3.2.5. The steps are:

i) Study on the different types of robot design that can be used for building inspection,

ii) Selection of The Type of Sensor to Be Used.

iii) Design of the selected robot for mobile robot navigation.

iv) Fabrication and assembly of robot.

v) Testing The Robot.

3.2.1 Study on the different types of robot design that can be used for building inspection

There are many types of robot design that exist today. The designs include:

i) Slithering mechanism such as a snake robot.

ii) Legged robot mechanism such as a robot dog.

iii) Wheeled type mobile robots.

These designs are mobile and not stationary such as a robot manipulator.

i. **Snake Robot**

Snake robots are designed to mimic the movement of real snakes, utilizing modular segments that allow for flexibility and manoeuvrability. They are typically composed of numerous individual joints, allowing them to slither through complex and confined spaces. Snake robots excel in navigating through tight areas and traversing uneven terrain, making them ideal for inspection and exploration tasks. However, a snake robot may not be the best fit for mobile robot navigation for building quality assessment. While snake robots can access hard-to-reach areas and inspect hidden spaces, their mobility may be limited when it comes to navigating traditional building structures, such as corridors, stairs, and elevators.

ii. **Robot Dog**

Robot dogs are robotic companions that resemble real dogs in appearance and movement. They are equipped with various sensors and actuators, allowing them to interact with the environment and perform a range of tasks. Robot dogs can exhibit locomotion, recognize objects, and even respond to commands. They are often used for assistance, surveillance, and entertainment purposes. In the context of my mobile robot, same as snake robot, a robot dog may not be the most suitable choice for mobile robot navigation for building quality assessment. While robot dogs can traverse through different terrains, their primary focus is on interaction and companionship rather than navigating complex indoor environments or assessing building quality.

iii. **Wheeled Robots**

Wheeled robot is a small-scale vehicle with wheels or tracks that can be operated remotely. These robots are typically equipped with cameras or sensors to provide real-time feedback to the operator. They are commonly used for exploration, surveillance, and remote inspection tasks. The focus on mobile robot navigation for building quality assessment, a remote-control car robot may be a more suitable option. With its manoeuvrability and ability to navigate indoor environments, wheeled robots can be controlled remotely to assess building quality effectively. Its small size allows it to move through narrow spaces, and its camera or sensor systems can capture detailed information about the building's condition.

3.2.1.1 Mechanism Design by using Computer Aided Design Software

i. Snake Robot

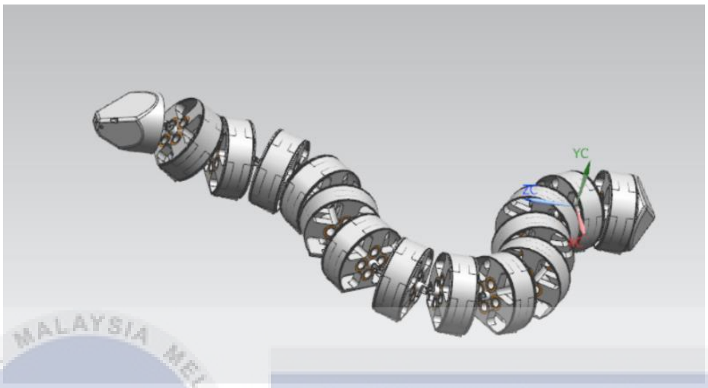


Figure 3.1: Snake Robot [54]

ii. Robot Dog

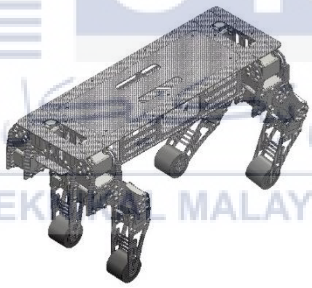


Figure 3.2: Snake Robot [55]

iii. Wheeled Robot

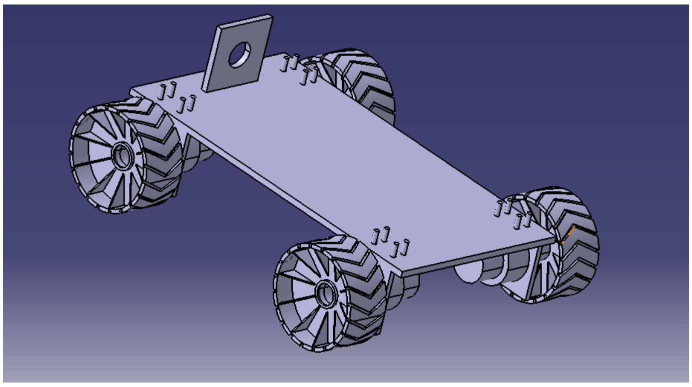


Figure 3.3: Wheeled Robot [56]

3.2.1.2 Compare Different Design Abilities

i. **Snake Robot**

The snake robot is designed with flexibility in mind, allowing it to navigate through narrow and complex spaces like pipes or rubble. Its segmented body and flexible joints enable it to slither and climb various surfaces, making it ideal for challenging terrains or search-and-rescue operations. Some snake robots also can manipulate objects or perform inspections in hard-to-reach locations, thanks to specialized end-effectors. Additionally, their modular design allows for customization and reconfiguration to adapt to different scenarios.

ii. **Robot Dog**

Robot dogs, on the other hand, prioritize mobility and stability. With their multiple legs, they can navigate rough terrain and traverse obstacles such as stairs or uneven surfaces. Balance and stability mechanisms enable them to maintain steady movement even during dynamic actions like running or jumping. These robots often feature sensors, cameras, and speakers for interaction with humans or other animals, providing companionship, assistance, or entertainment. Some robot dogs are also autonomous, capable of navigating and performing tasks without constant human control.

iii. **Wheeled Robot**

Wheeled robots excel in speed, agility, and versatility. Designed for high-speed movements and quick turns, they are often used for racing, exploration, or surveillance. Depending on their design, they can be tailored to various terrains, including off-road or indoor environments, and may incorporate features like shock absorbers, all-wheel drive, or specialized tires. Wheeled robots can also carry and transport objects, making them useful for delivery or logistics purposes. Their modular design allows for customization and upgrades, such as adding additional sensors, cameras, or manipulator arms.

3.2.2 Selection of The Type of Sensor to Be Used

Ultrasonic sensors are perfect for my project titled "Mobile Robot Navigation for Building Quality Assessment". One of the main reasons why time of flight and ultrasonic sensors are the best sensor for this application is their ability to accurately measure distances and detect obstacles in real time.

This feature is very important when mobile robots are moved indoors and building quality needs to be assessed. ToF (time of flight) sensors generally achieve higher accuracy compared to ultrasonic sensors because they utilize light signals, which travel at a faster pace than the sound waves employed by ultrasonic sensors. These sensors gauge the round-trip time for light to reach a target and return, delivering precise and real-time distance measurements with enhanced resolution. Moreover, ToF sensors frequently exhibit an extended detection range, possess a more compact design, and deliver superior performance in applications where exact distance measurements are essential. Meanwhile, ultrasonic sensors emit high-frequency sound waves and measure the time it takes for the sound waves to be reflected after they hit an object. This will allow the sensor to calculate the distance between the robot and the obstacle. When assessing building quality, it is important to accurately determine the distance between the robot and walls, furniture, and other structural elements.

Moreover, ultrasonic sensors offer a wide detection range, typically covering several meters, making them well-suited for evaluating building quality in diverse environments. When integrated with time-of-flight (ToF) sensors, which capitalize on the swift travel of light signals to provide accurate and real-time distance measurements, this combination enhances the assessment process. Together, these sensors enable comprehensive evaluations of structures, allowing for the observation of objects at different heights and angles and offering a nuanced understanding of a building's overall integrity and potential flaws. Additionally, the reliable performance of ultrasonic sensors in varying lighting conditions complements the precision and compact design of ToF sensors, creating a robust solution for assessing building quality across a variety of scenarios.

Unlike optical sensors, which have difficulty in dim or highly reflective environments, ultrasonic sensors work regardless of lighting conditions. This enables mobile robots to accurately navigate and assess building quality regardless of ambient lighting conditions.

Additionally, while ultrasonic sensors offer cost-effective and readily available options, I also leverage time-of-flight (ToF) sensors in my projects for enhanced accuracy and performance. Despite being more economical, ultrasonic sensors prove practical for integration into robotic systems and demonstrate compatibility with a diverse range of microcontrollers and platforms. This affordability and accessibility have established ultrasonic sensors as a favoured choice in the realm of robotics and automation applications. On the other hand, the application of ToF sensors in my projects adds an extra layer of precision, benefiting from their capacity to measure distances with higher accuracy and their suitability for scenarios where the speed of light is advantageous. This combination allows for a versatile and effective approach to distance sensing in various project implementations.

3.2.3 Design of the selected robot for mobile robot navigation

In this section, the robot that will perform navigation will be designed. After selecting the design of robot from the previous comparison, the size, length, and shape of the robot need to be determined. Also, the actuators (motors), couplings, body material etc need to be selected.

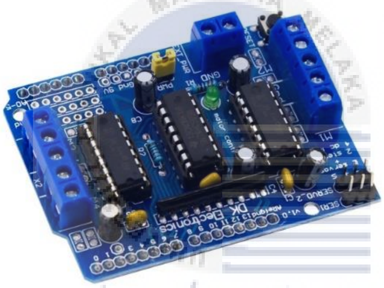
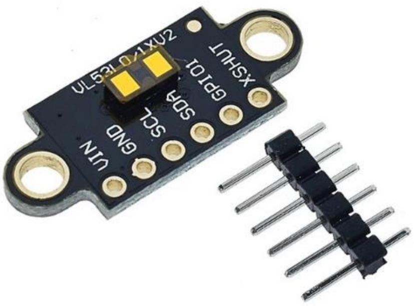
The design will be done in AutoCAD / SolidWorks / Fusion 360 and basic tests in simulation will be conducted, such as stress test.

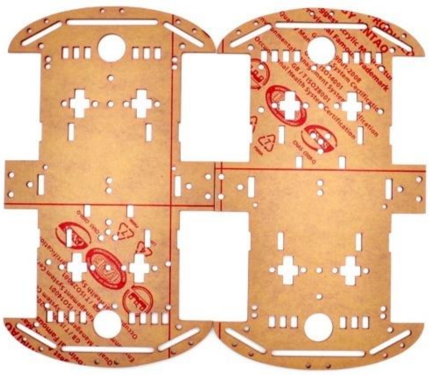
3.2.4 Fabrication and assembly of robot

Fabrication and assembly of robot involves the process of creating and building robotic systems. The manufacturing industry is the manufacture of various parts and parts that make up robots, such as robot body structures, mechanical parts, and electronic circuits. This often involves cutting, shaping, and joining materials such as metals, plastics, and composites to create the desired robot design. Assembly, on the other hand, is the stage of connecting and integrating these manufactured parts to form a fully functional robot. In this case, with buy online the materials for creating and assembling your robot, as designs for different types of robots are on sale.

3.2.5 Component of The Mobile Robot Navigation

Table 3.1: Component of The Mobile Robot Navigation

Component	Units	Reason
Arduino UNO 	1	-Interface for programming and connecting various sensors and motors.
L293D Motor Drive Shield 	1	-Motor control and drive for the DC motor of the robot. -Enables bidirectional control of up to four DC motors.
Ultrasonic 	1	- To measure the distance to the wall to avoid obstacles. -Can detect distance up to more than 2 meters.
Time of Flight 	2	-Provides accurate distance measurement for navigation mobile robot. -This sensor suitable for mobile robot to follow the wall.

4WD Robot Chassis Kit 	1	-Designed for 4-wheel drive, enhancing stability and maneuverability. -For platform assembling mobile robot.
Ultrasonic Holder 	1	-Useful for ultrasonic to stable distance sensing.
Battery Holder 	1	-Holds and organizes the power source. -Easy to use for replacement batteries.
Battery 3.7V 	2	-Suitable voltage for the motor drive and other electronics.

DC Motor 	4	-Drive the wheels for robot movement and control.
Mecanum Wheels 	4	-Enhances the robot’s ability to move in any direction with ease. -Provides omnidirectional movement, like allowing for versatile mobility.

3.2.6 Schematic Diagram of Mobile Robot Navigation

Fritzing was used to draw and design the circuit electrical drawing to easily create and share electronics circuits.

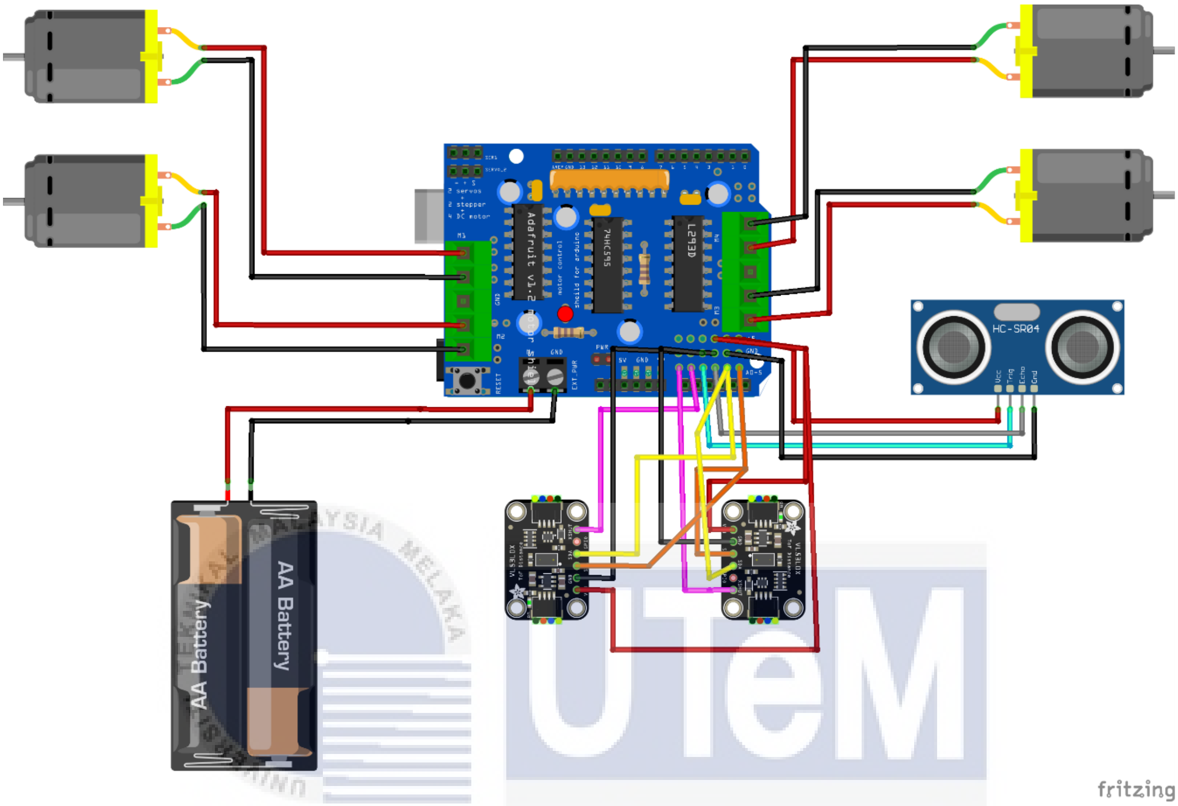


Figure 3.4: Schematic Diagram Mobile Robot Navigation

3.2.7 Testing The Robot

The main reason for choosing a wheeled robot design is its simple movement mechanism such as exploration, surveillance, and obstacles avoidance. Wheeled robots can use different movement methods depending on the type of wheel they use where to test the efficiency of the wheel to perform the movement robot. As in the figure below that we illustrate, the use of mecanum wheels is more suitable for wheeled robot movement because that type of wheel is easy to move.

i. The Movement Direction Robot

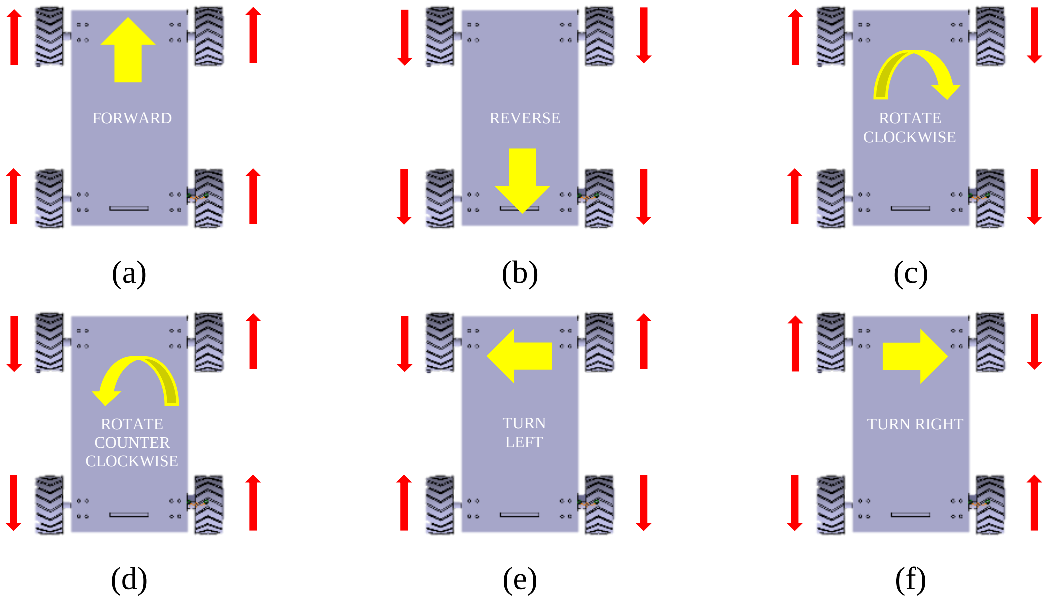


Figure 3.5: Movement Robot

As seen in Figure 3.4, the Mecanum wheel designed robot is tested for different moving directions. This includes (a) Robot moving in forward direction, (b) Robot moving in reverse direction, (c) Robot moving in rotate clockwise, (d) Robot moving in rotate counterclockwise, (e) Robot moving in left direction, and (f) Robot moving in right direction.

ii. Detection of Wall by using Sensors

This methodology is needed to ensure that the wall is successfully detected by the robot. The robot has installed two time of flight sensors and one ultrasonic sensor where two sensors are installed in front and one on the left side of the robot. The illustration is shown in Figure 3.5.

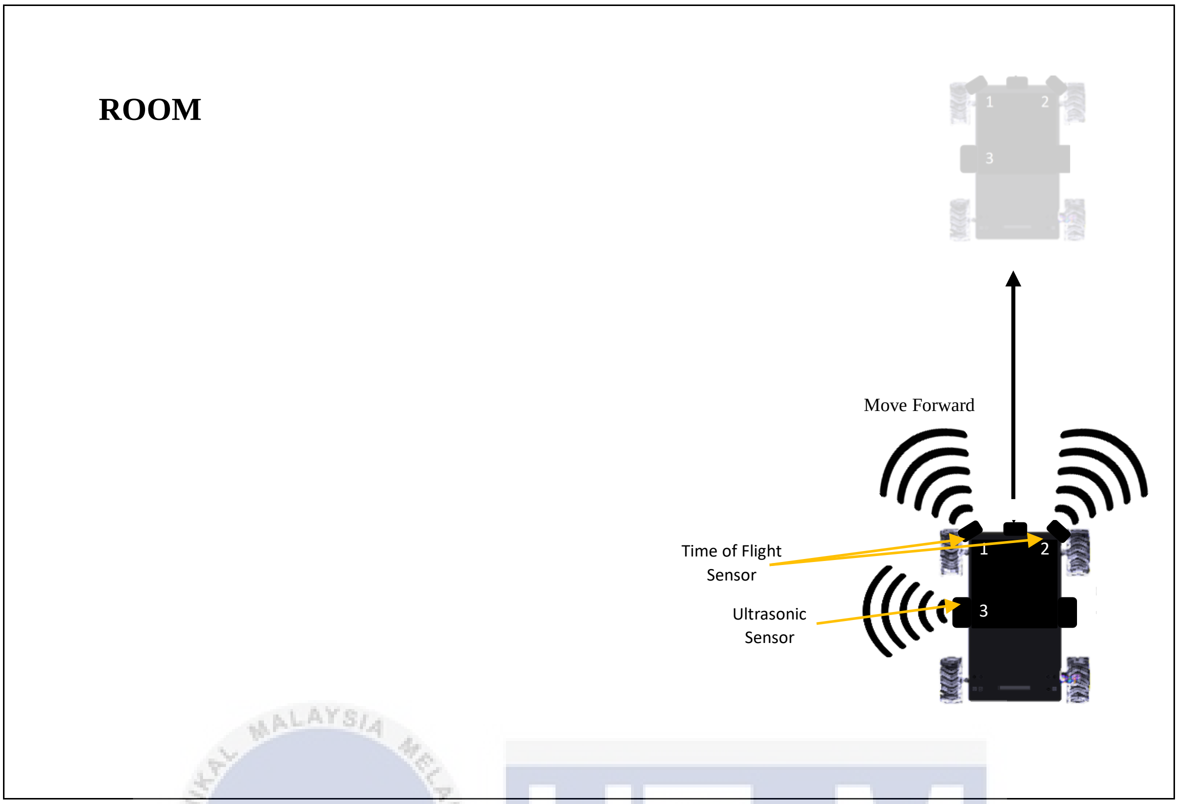


Figure 3.6: Robot Move Forward

In Figure 3.5, the robot has multiple sensors attached. Wheeled robot will move forward until the sensor can detect the wall with a predetermined or modified distance to the sensor.

3.3 Algorithm of wall following robot

This methodology is related to objective 2. For objective 2 need do the flowchart by referred in objective 3. There have two type of designs dimension that popular in Malaysia such as rectangle shape and L-shape below.

3.3.1 Rectangle Shape Movement Robot and Flowchart

A rectangular shape room's efficient use of space, unobstructed lines of sight, and ease of mobility make it a great environment for movement robots to survey construction sites. Because of its uniform design, the robot can move around with ease, avoiding obstructions and providing the most coverage. The robot using five sensors can precisely detect and map the surroundings with straight walls and corners, making it possible to effectively evaluate the building's structural stability.

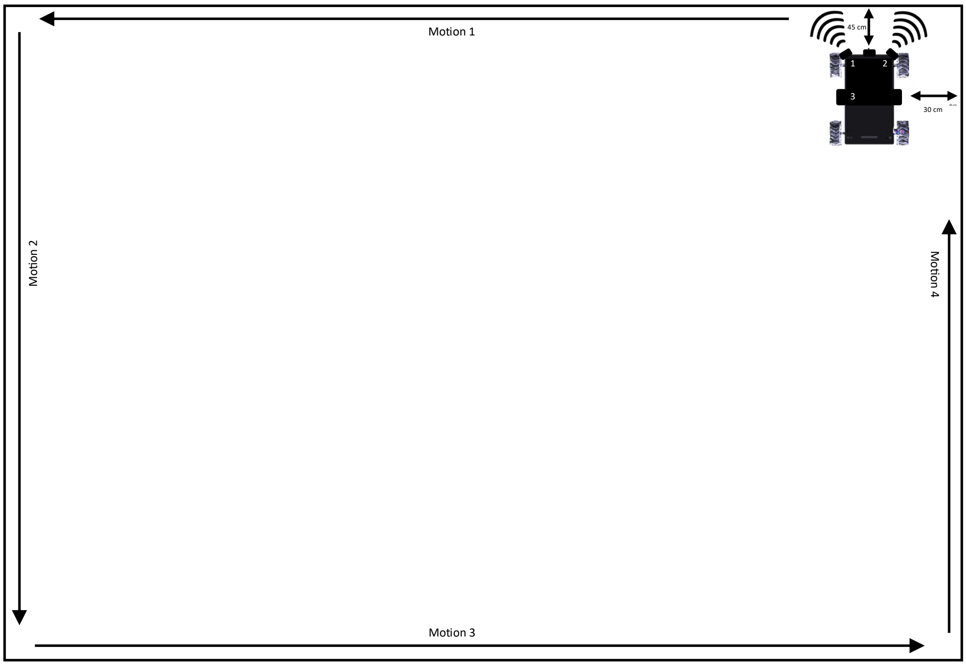


Figure 3.7: Rectangle Shape

In Figure 3.6, the sensor on the mobile robot detects the wall in the rectangle shape room where the robot starts moving on the left side until it completes one rotation of room. Sensors 1, 2 will detect the wall with a distance of 45 cm while sensors 3 will detect a distance of 45 cm. This robot will complete all tasks from motion 1 to motion 4.

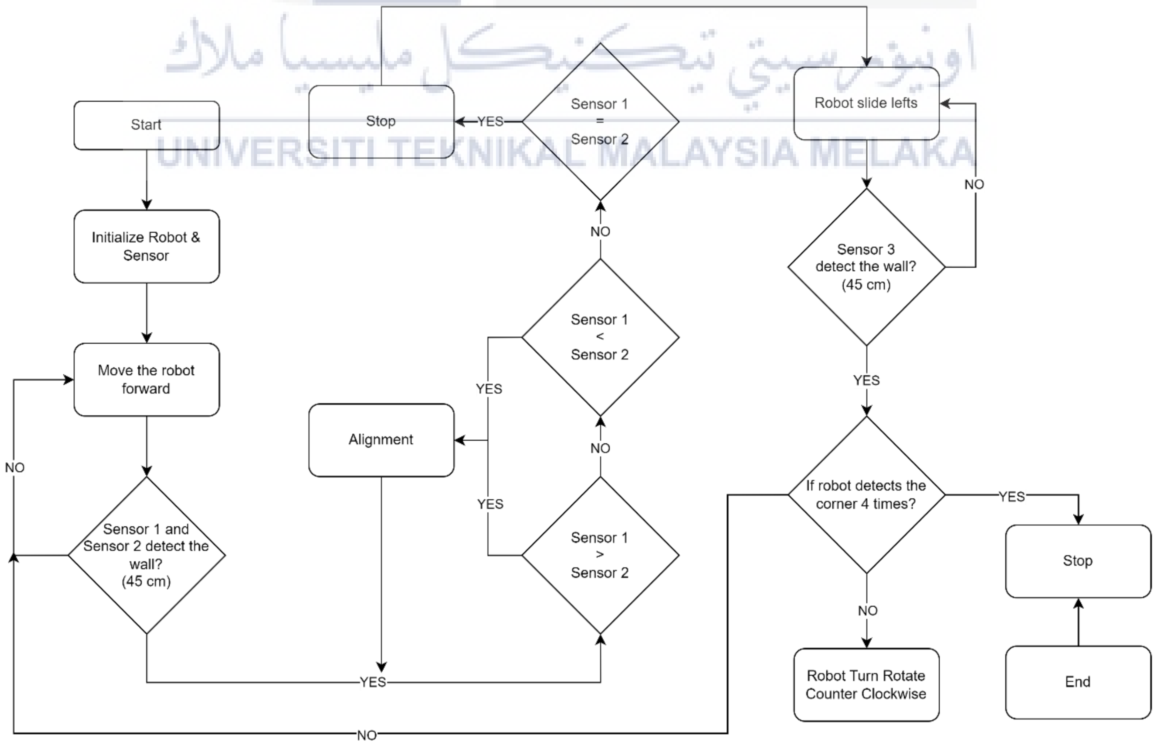


Figure 3.8: Flowchart Movement Robot in Rectangle Room

Figure 3.7 shows the flowchart of the rectangular shape room. The algorithm starts from initialize robot and sensor where the robot will be placed in the room to be tested. Next, the robot will starts moving forward, sensor 1 and sensor 2 will detect the wall in front and the sensor on the robot will stop at a distance of 45 cm from the wall. When the distance position of sensor 1 is more than sensor 2 and position of sensor 2 is more than sensor 1, then the robot will make an alignment. And, if both the distance of sensor 2 and sensor 3 are the distance are same, then the robot will stop at the exact position between the wall and the robot will move to the left to complete the motion 1 until sensor 4 will detect the wall on the left with a distance of 45 cm from wall. After the sensor 3 can detect the wall, the robot will make a counterclockwise rotation and will do the work for motion 2, motion 3 and motion 4.

3.3.2 L-Shape Movement Robot and Flowchart

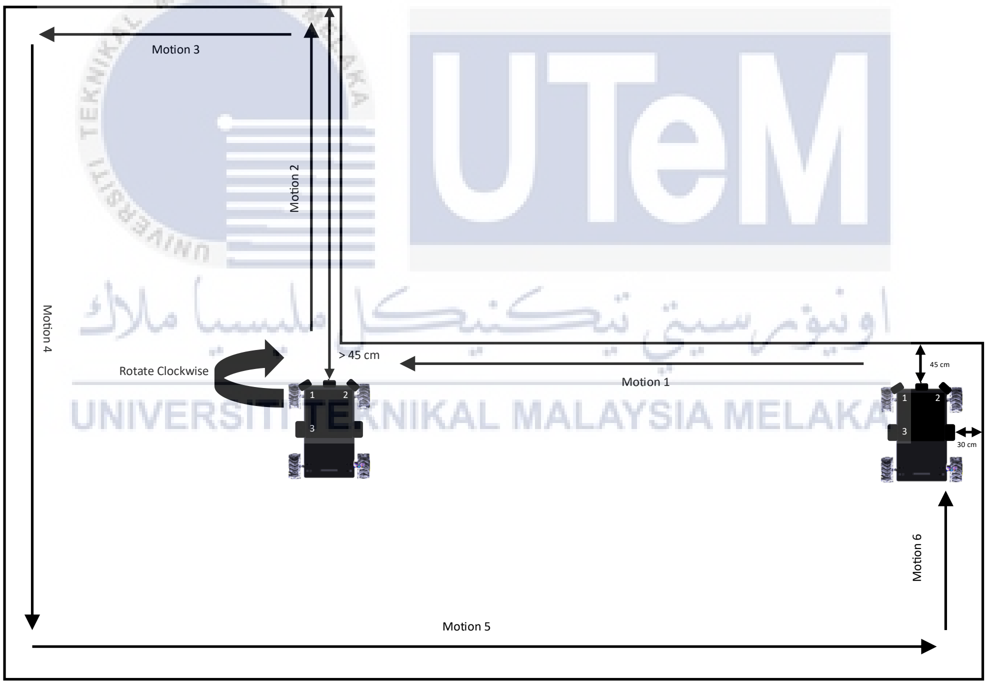


Figure 3.9: L-Shape Room

In Figure 3.7, the sensor on the mobile robot detects the wall in the L-shape room where the robot starts moving on the left side until it completes one rotation of room but has a difference in robot movement due to the shape of the room. Sensors 1 and 2 will detect the wall with a distance of 45 cm while sensors 3 will detect a distance of 45 cm same tasks as rectangle shape room. This robot will complete all tasks from motion 1 to motion 6.

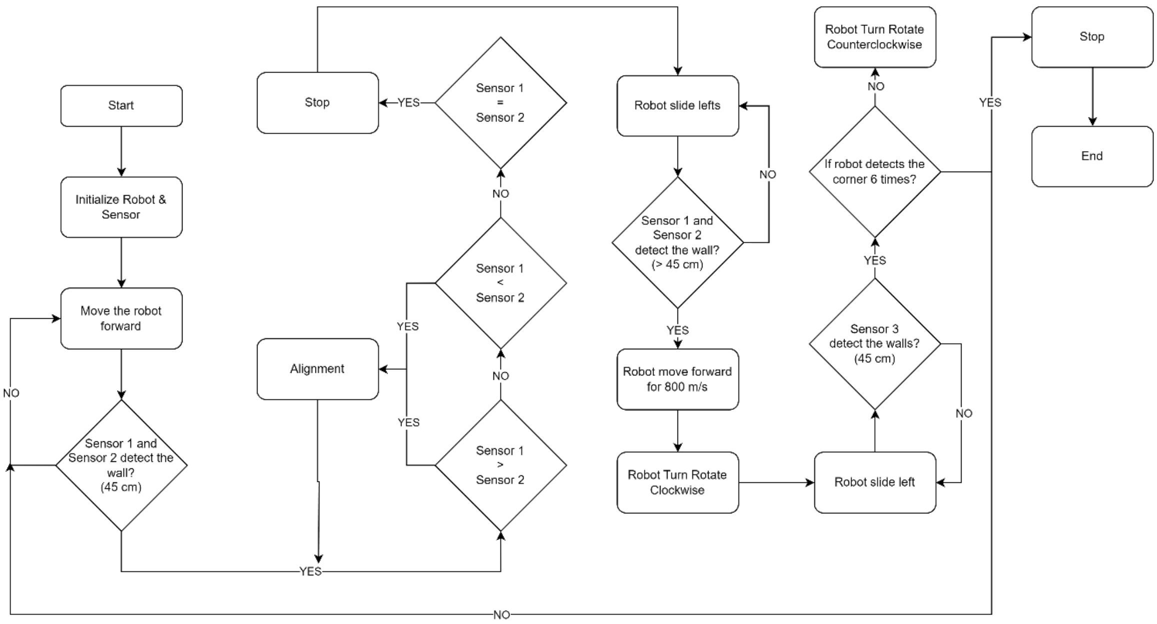


Figure 3.10: Flowchart Movement Robot in L-Shape Room

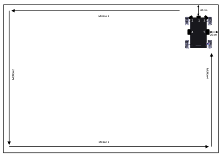
Figure 3.8 shows the flowchart of L-shape room. The algorithm starts from initialize robot and sensor where the robot will be placed in the room to be tested. Next, the robot will starts moving forward, sensor 1 and sensor 2 will detect the wall in front and the sensor on the robot will stop at a distance of 45 cm from the wall. When the distance position of sensor 1 is more than sensor 2 and position of sensor 2 is more than sensor 1, then the robot will make an alignment. And, if both the distance of sensor 1 and sensor 2 are the distance are same, then the robot will stop at the exact position between the wall and the robot will move to the left to complete the motion 1. However, if Sensor 1 and sensor 2 not detects the wall, the robot will move forward for 0.8 seconds, and make clockwise direction and move forward again until sensor detect the wall. Then, the robot will move forward until Sensor 1 and Sensor 2 detect the wall at a distance of 45 cm and it’s detected the wall, the robot will continue to move to the left and complete all movements from motion 1 to motion 6.

3.4 Performance Movement Robot

This methodology is related to objective 3. For objective 3 need to evaluate the performance of robot following the wall referred in objective 2. For this part it is necessary to record data from the movement of the robot in the form of the time taken for the robot to complete the task for one round in the room.

3.4.1 Theoretical Data Movement Mobile Robot

Table 3.2: Data Calculation Movement Robot Rectangle Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
Motion 1 (4 m)	8 times	8.125 seconds	400 ms
Motion 2 (3.5 m)	7 times	6.5 seconds	
Motion 3 (4 m)	8 times	8.125 seconds	
Motion 4 (3.5 m)	7 times	6.5 seconds	

Dimensions of room,

Length x Width = 4-meter x 3.5-meters

Time of Flight Sensor detects distance = 45 cm

Ultrasonic Sensor detects distance = 45 cm

Distance length robot slide left for LENGTH, 4 m – (0.3 + 0.45) m = 3.25 m

Distance length robot slide left for WIDTH, 3.5 m – (0.45 + 0.45) m = 2.6 m

Time to complete,

Motion 1 = $\frac{Distance}{Speed} = \frac{3.25\ m}{0.4\ s} = 8.125\ seconds$

Motion 2 = $\frac{Distance}{Speed} = \frac{2.6\ m}{0.4\ s} = 6.5\ seconds$

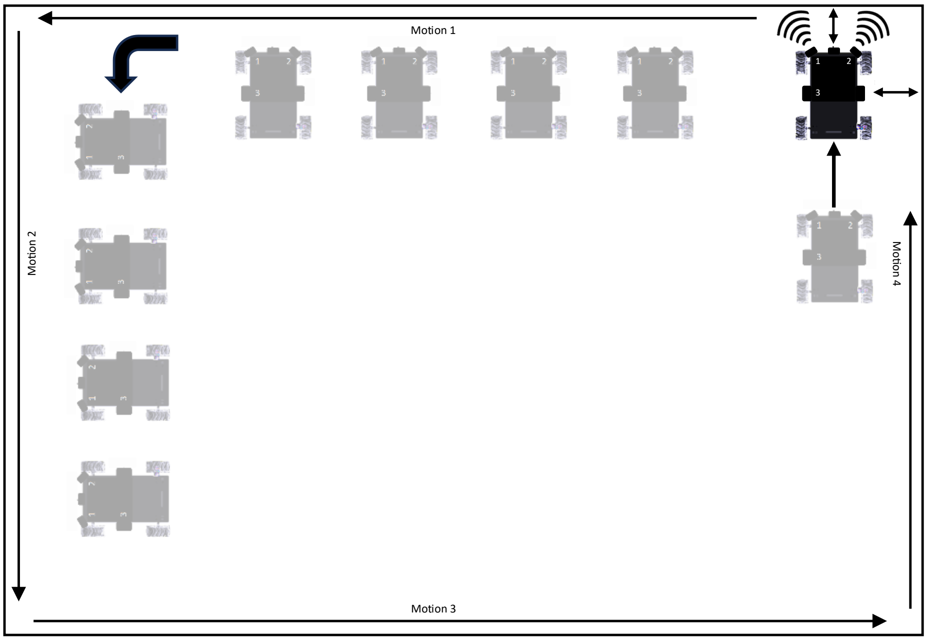
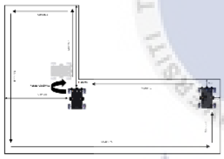


Figure 3.11: Mobile Robot Slide Left in Rectangle Shape Room

Table 3.3: Data Calculation Movement Robot for L-Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
Motion 1 (2.5 m)	6 times	5.5 seconds	
Motion 2 (1.25 m)	3 times	2 seconds	
Motion 3 (1.5 m)	2 times	1.5 seconds	400 ms
Motion 4 (3.5 m)	7 times	6.5 seconds	
Motion 5 (4 m)	8 times	8.125 seconds	
Motion 6 (1.25 m)	3 times	2 seconds	

Dimensions of room,

Length x Width = 4-meter x 3.5-meters

L-shape dimensions Length = 2.5-meter (Motion 1) x 1.25-meters (Motion 2) x 1.5-meters (Motion 3) x 3.5-meters (Motion 4) x 4-meters (Motion 5) x 1.25-meters (Motion 6)

Time of Flight Sensor detects distance = 45 cm

Ultrasonic Sensor detects distance = 45 cm

Distance length robot slide left for LENGTH L-SHAPE, 2.5 m – 0.3 m = 2.2 m

Distance length robot slide left for WIDTH L-SHAPE, 1.25 m – 0.45 m = 0.8 m

Distance length robot slide left for MOTION 3, 1.5 m – (0.45 + 0.45) m = 0.6 m

Time to complete,

Motion 1 = $\frac{Distance}{Speed} = \frac{2.2\ m}{0.4\ s} = 5.5\ seconds$

Motion 2 = $\frac{Distance}{Speed} = \frac{0.8\ m}{0.4\ s} = 2\ seconds$

Motion 3 = $\frac{Distance}{Speed} = \frac{0.6\ m}{0.4\ s} = 1.5\ seconds$

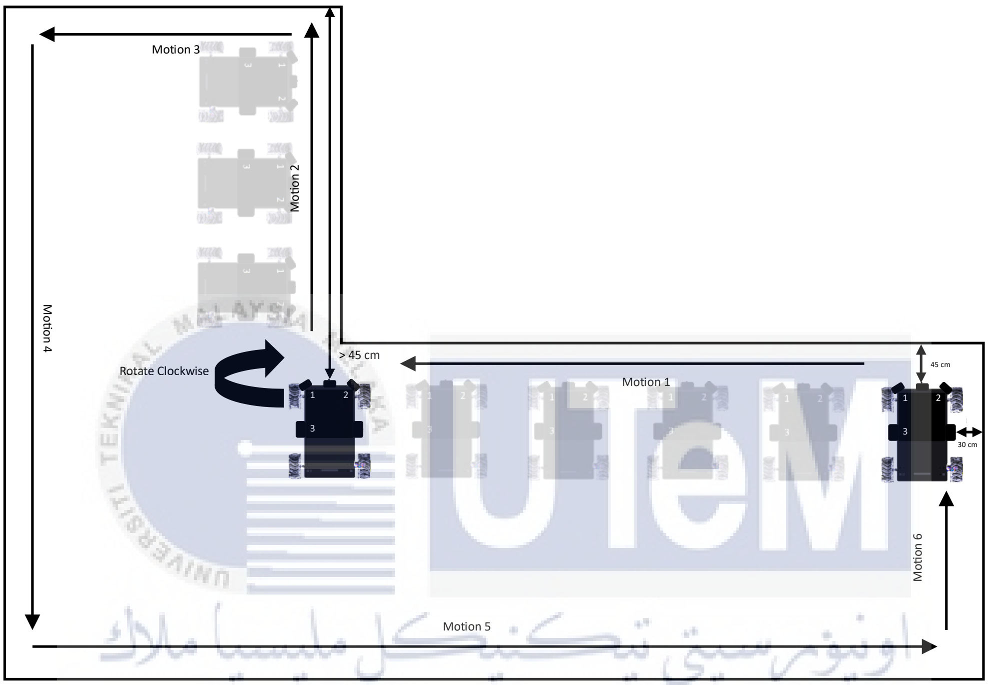


Figure 3.12: Mobile Robot Slide Left in L-Shape Room

For this part, there are two tables above showing Data Calculation Movement Robot. there are four motions for Rectangle Room and five motions for L-Shape Room. Each dimension needs to record data for the robot to complete one motion in second with the total of the slide left movement robot.

In this Figure 3.10 and Figure 3.11 shown how mobile robot move starting from right corner. If mobile robot moves forward to the wall for 2 seconds, robot will stop. Distance from right robot with wall, let’s say in 30 cm and in front of robot in 45 cm. The robot will move slide left until the next wall and will perform alignment if any sensor (sensor 1 or sensor 2) is unstable.

CHAPTER 4

RESULT

4.1 Introduction

In this project, we design a mobile robot capable of autonomously navigating through added to architectural contexts in order to improve the accuracy and efficiency of building quality evaluation. We aim to develop an intelligent system that integrates real-time data collection, complex algorithms, and precise mapping techniques to discover and analyse numerous structural and aesthetic parameter. With the ability to enable quicker decision-making, lower the risk of human error, and ultimately guarantee the delivery of greater building quality, this innovative method has the potential to greatly enhance the evaluation process.

4.2 Drawing Result and Final Product

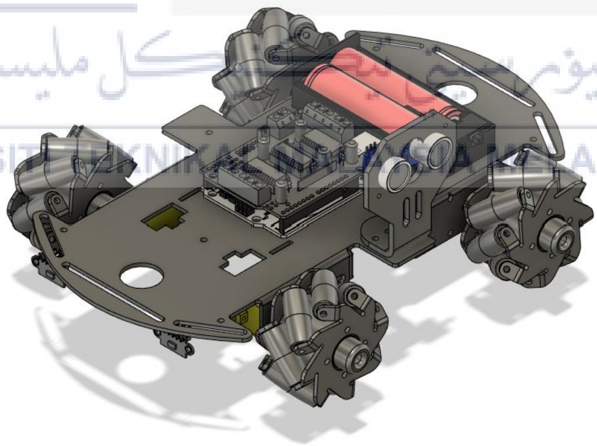


Figure 4.1:Final Drawing Design Mobile Robot

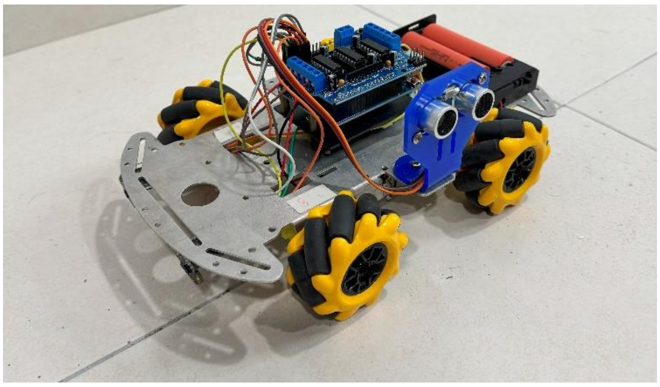


Figure 4.2: Final Product Mobile Robot (Isometric View)

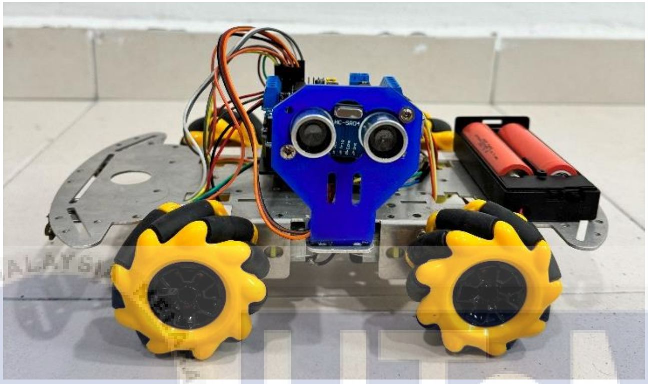


Figure 4.3: Final Product Mobile Robot (Left View)

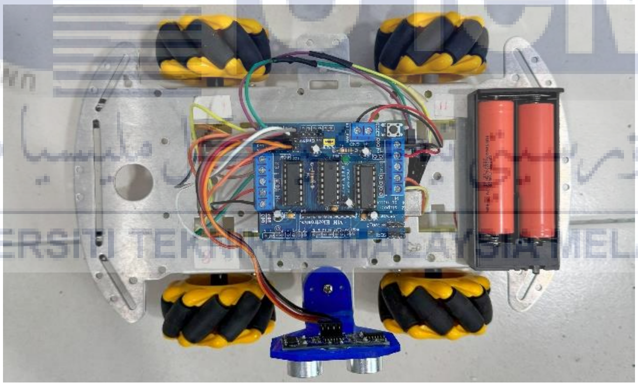


Figure 4.4: Final Product Mobile Robot (Top View)

The initial final drawing above was designed in the Fusion 360 drawing software that shows the initial impression of the mobile robot project and final product that followed by final drawing. One ultrasonic sensor has been installed on the robot and two time of flight sensors are installed below from chassis robot along with four mecanum wheels that are installed on the DC motor that is under the robot chassis.

4.3 Experiment Data Movement Mobile Robot

Mobile Robot Movement Experimental Data is the data obtained after the mobile robot performed a manual simulation of two types of rooms namely Rectangle Shape Room and L-Shape Room. There are three types of dimensions which are in Room A, Room B and Room C which have different dimensions and have mobile robot sliding delay time of 400 ms and 800 ms which are experimented which is the end of the data provided below.

4.3.1 Room A (Rectangle Shape Room)

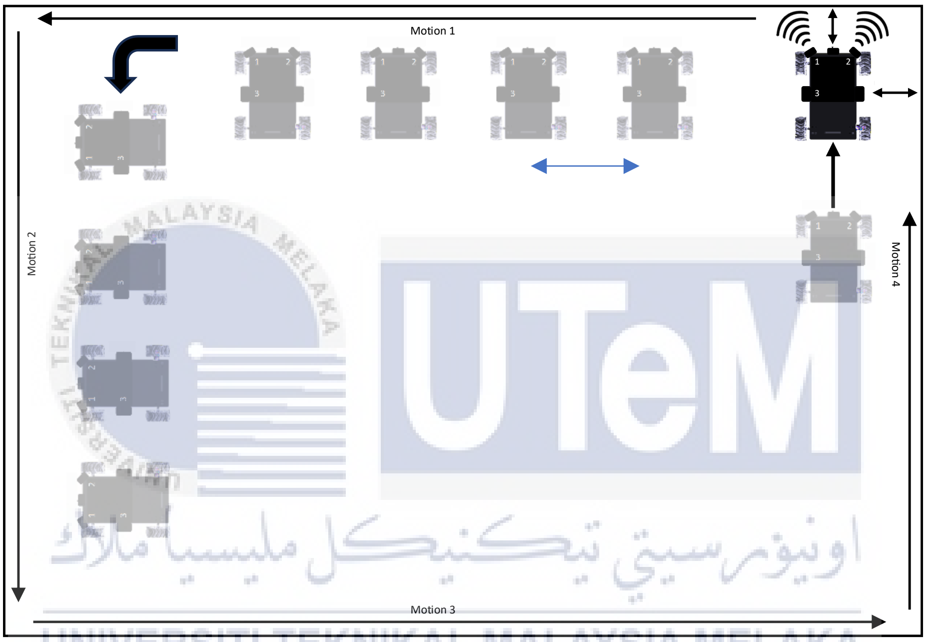


Figure 4.5: Mobile Robot Slide Left in Rectangle Shape Room

Dimensions of room A,

Length x Width = 3.18-meter x 3.20-meters

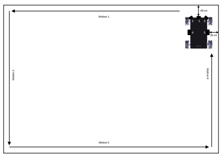
Time of Flight Sensor detects distance = 45 cm

Ultrasonic Sensor detects distance = 45 cm

Distance length robot slide left for LENGTH, 3.18 m – (0.3 + 0.45) m = 2.48 m

Distance length robot slide left for WIDTH, 3.2 m – (0.45 + 0.45) m = 2.3 m

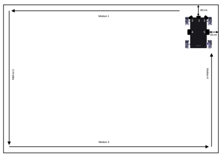
Table 4.1: Data Calculation Movement Robot Rectangle Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
		1st Data	
Motion 1 (3.18 m)	18 times	21 seconds	
Motion 2 (3.20 m)	16 times	25 seconds	
Motion 3 (3.18 m)	16 times	24 seconds	400 ms
Motion 4 (3.20 m)	17 times	25 seconds	
Total	67 times	95 seconds	
		2nd Data	
Motion 1 (3.18 m)	20 times	25 seconds	
Motion 2 (3.20 m)	18 times	24 seconds	
Motion 3 (3.18 m)	16 times	23 seconds	400 ms
Motion 4 (3.20 m)	17 times	24 seconds	
Total	71 times	96 seconds	
		3rd Data	
Motion 1 (3.18 m)	19 times	24 seconds	
Motion 2 (3.20 m)	17 times	24 seconds	
Motion 3 (3.18 m)	16 times	22 seconds	400 ms
Motion 4 (3.20 m)	17 times	23 seconds	
Total	69 times	93 seconds	
		4th Data	
Motion 1 (3.18 m)	19 times	24 seconds	
Motion 2 (3.20 m)	17 times	22 seconds	
Motion 3 (3.18 m)	16 times	23 seconds	400 ms
Motion 4 (3.20 m)	17 times	23 seconds	
Total	69 times	92 seconds	

		5th Data	
Motion 1 (3.18 m)	21 times	25 seconds	
Motion 2 (3.20 m)	18 times	23 seconds	
Motion 3 (3.18 m)	17 times	23 seconds	400 ms
Motion 4 (3.20 m)	17 times	22 seconds	
Total	73 times	93 seconds	
		6th Data	
Motion 1 (3.18 m)	20 times	25 seconds	
Motion 2 (3.20 m)	18 times	23 seconds	
Motion 3 (3.18 m)	16 times	24 seconds	400 ms
Motion 4 (3.20 m)	17 times	24 seconds	
Total	71 times	96 seconds	
		7th Data	
Motion 1 (3.18 m)	20 times	24 seconds	
Motion 2 (3.20 m)	17 times	23 seconds	
Motion 3 (3.18 m)	16 times	23 seconds	400 ms
Motion 4 (3.20 m)	17 times	22 seconds	
Total	70 times	94 seconds	
		8th Data	
Motion 1 (3.18 m)	19 times	23 seconds	
Motion 2 (3.20 m)	18 times	22 seconds	
Motion 3 (3.18 m)	16 times	23 seconds	400 ms
Motion 4 (3.20 m)	17 times	24 seconds	
Total	70 times	93 seconds	
		9th Data	
Motion 1 (3.18 m)	20 times	25 seconds	
Motion 2 (3.20 m)	18 times	23 seconds	
Motion 3 (3.18 m)	16 times	21 seconds	400 ms
Motion 4 (3.20 m)	17 times	22 seconds	
Total	71 times	96 seconds	

10th Data			
Motion 1 (3.18 m)	21 times	25 seconds	400 ms
Motion 2 (3.20 m)	19 times	24 seconds	
Motion 3 (3.18 m)	17 times	23 seconds	
Motion 4 (3.20 m)	17 times	24 seconds	
Total	74 times	94 seconds	

Table 4.2: Data Calculation Movement Robot Rectangle Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
1st Data			
Motion 1 (3.18 m)	15 times	19 seconds	800 ms
Motion 2 (3.20 m)	13 times	23 seconds	
Motion 3 (3.18 m)	13 times	21 seconds	
Motion 4 (3.20 m)	14 times	23 seconds	
Total	55 times	86 seconds	
2nd Data			
Motion 1 (3.18 m)	16 times	21 seconds	800 ms
Motion 2 (3.20 m)	14 times	22 seconds	
Motion 3 (3.18 m)	12 times	20 seconds	
Motion 4 (3.20 m)	13 times	21 seconds	
Total	55 times	84 seconds	
3rd Data			
Motion 1 (3.18 m)	15 times	20 seconds	800 ms
Motion 2 (3.20 m)	13 times	20 seconds	
Motion 3 (3.18 m)	12 times	18 seconds	
Motion 4 (3.20 m)	13 times	19 seconds	
Total	53 times	77 seconds	

4th Data			
Motion 1 (3.18 m)	15 times	20 seconds	
Motion 2 (3.20 m)	13 times	18 seconds	
Motion 3 (3.18 m)	12 times	19 seconds	800 ms
Motion 4 (3.20 m)	13 times	19 seconds	
Total	53 times	76 seconds	
5th Data			
Motion 1 (3.18 m)	17 times	21 seconds	
Motion 2 (3.20 m)	14 times	19 seconds	
Motion 3 (3.18 m)	13 times	19 seconds	800 ms
Motion 4 (3.20 m)	13 times	18 seconds	
Total	57 times	77 seconds	
6th Data			
Motion 1 (3.18 m)	16 times	21 seconds	
Motion 2 (3.20 m)	14 times	19 seconds	
Motion 3 (3.18 m)	12 times	20 seconds	800 ms
Motion 4 (3.20 m)	13 times	20 seconds	
Total	55 times	80 seconds	
7th Data			
Motion 1 (3.18 m)	16 times	20 seconds	
Motion 2 (3.20 m)	13 times	19 seconds	
Motion 3 (3.18 m)	12 times	19 seconds	800 ms
Motion 4 (3.20 m)	13 times	18 seconds	
Total	54 times	76 seconds	
8th Data			
Motion 1 (3.18 m)	15 times	19 seconds	
Motion 2 (3.20 m)	14 times	18 seconds	
Motion 3 (3.18 m)	12 times	19 seconds	800 ms
Motion 4 (3.20 m)	13 times	20 seconds	
Total	54 times	76 seconds	

9th Data			
Motion 1 (3.18 m)	16 times	21 seconds	800 ms
Motion 2 (3.20 m)	14 times	19 seconds	
Motion 3 (3.18 m)	12 times	17 seconds	
Motion 4 (3.20 m)	13 times	18 seconds	
Total	55 times	75 seconds	
10th Data			
Motion 1 (3.18 m)	17 times	21 seconds	800 ms
Motion 2 (3.20 m)	15 times	20 seconds	
Motion 3 (3.18 m)	13 times	19 seconds	
Motion 4 (3.20 m)	13 times	20 seconds	
Total	58 times	80 seconds	

4.3.2 Room B (Rectangle Shape Room)

Dimensions of room B,

Length x Width = 2.52-meter x 3.10-meters

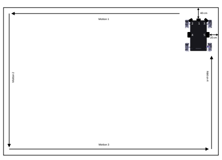
Time of Flight Sensor detects distance = 45 cm

Ultrasonic Sensor detects distance = 45 cm

Distance length robot slide left for LENGTH, 2.52 m – (0.3 + 0.45) m = 1.77 m

Distance length robot slide left for WIDTH, 3.10 m – (0.45 + 0.45) m = 2.2 m

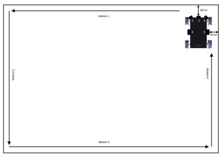
Table 4.3: Data Calculation Movement Robot Rectangle Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
1st Data			
Motion 1 (2.52 m)	14 times	16 seconds	400 ms
Motion 2 (3.10 m)	13 times	19 seconds	
Motion 3 (2.52 m)	13 times	18 seconds	
Motion 4 (3.10 m)	13 times	19 seconds	

Total	53 times	72 seconds	
2nd Data			
Motion 1 (2.52 m)	16 times	19 seconds	400 ms
Motion 2 (3.10 m)	14 times	18 seconds	
Motion 3 (2.52 m)	13 times	17 seconds	
Motion 4 (3.10 m)	14 times	18 seconds	
Total	56 times	72 seconds	
3rd Data			
Motion 1 (2.52 m)	15 times	18 seconds	400 ms
Motion 2 (3.10 m)	13 times	18 seconds	
Motion 3 (2.52 m)	13 times	16 seconds	
Motion 4 (3.10 m)	14 times	17 seconds	
Total	55 times	69 seconds	
4th Data			
Motion 1 (2.52 m)	15 times	18 seconds	400 ms
Motion 2 (3.10 m)	13 times	16 seconds	
Motion 3 (2.52 m)	13 times	17 seconds	
Motion 4 (3.10 m)	14 times	17 seconds	
Total	55 times	68 seconds	
5th Data			
Motion 1 (2.52 m)	16 times	19 seconds	400 ms
Motion 2 (3.10 m)	14 times	17 seconds	
Motion 3 (2.52 m)	13 times	17 seconds	
Motion 4 (3.10 m)	13 times	16 seconds	
Total	57 times	69 seconds	
6th Data			
Motion 1 (2.52 m)	16 times	19 seconds	400 ms
Motion 2 (3.10 m)	14 times	17 seconds	
Motion 3 (2.52 m)	13 times	18 seconds	
Motion 4 (3.10 m)	14 times	18 seconds	
Total	56 times	72 seconds	

	7th Data		
Motion 1 (2.52 m)	16 times	18 seconds	
Motion 2 (3.10 m)	13 times	17 seconds	
Motion 3 (2.52 m)	13 times	17 seconds	400 ms
Motion 4 (3.10 m)	14 times	16 seconds	
Total	55 times	68 seconds	
	8th Data		
Motion 1 (2.52 m)	14 times	16 seconds	
Motion 2 (3.10 m)	13 times	15 seconds	
Motion 3 (2.52 m)	13 times	16 seconds	400 ms
Motion 4 (3.10 m)	14 times	17 seconds	
Total	53 times	64 seconds	
	9th Data		
Motion 1 (2.52 m)	16 times	19 seconds	
Motion 2 (3.10 m)	14 times	17 seconds	
Motion 3 (2.52 m)	13 times	15 seconds	400 ms
Motion 4 (3.10 m)	14 times	16 seconds	
Total	56 times	72 seconds	
	10th Data		
Motion 1 (2.52 m)	16 times	19 seconds	
Motion 2 (3.10 m)	15 times	18 seconds	
Motion 3 (2.52 m)	13 times	17 seconds	400 ms
Motion 4 (3.10 m)	13 times	18 seconds	
Total	57 times	72 seconds	

Table 4.4: Data Calculation Movement Robot Rectangle Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
		1st Data	
Motion 1 (2.52 m)	12 times	14 seconds	
Motion 2 (3.10 m)	11 times	16 seconds	
Motion 3 (2.52 m)	13 times	15 seconds	800 ms
Motion 4 (3.10 m)	11 times	16 seconds	
Total	47 times	61 seconds	
		2nd Data	
Motion 1 (2.52 m)	15 times	16 seconds	
Motion 2 (3.10 m)	13 times	15 seconds	
Motion 3 (2.52 m)	12 times	14 seconds	800 ms
Motion 4 (3.10 m)	12 times	15 seconds	
Total	52 times	60 seconds	
		3rd Data	
Motion 1 (2.52 m)	12 times	15 seconds	
Motion 2 (3.10 m)	12 times	15 seconds	
Motion 3 (2.52 m)	11 times	13 seconds	800 ms
Motion 4 (3.10 m)	13 times	14 seconds	
Total	48 times	57 seconds	
		4th Data	
Motion 1 (2.52 m)	13 times	15 seconds	
Motion 2 (3.10 m)	12 times	13 seconds	
Motion 3 (2.52 m)	11times	14 seconds	800 ms
Motion 4 (3.10 m)	12 times	14 seconds	
Total	48 times	56 seconds	

5^{th} Data			
Motion 1 (2.52 m)	13 times	16 seconds	800 ms
Motion 2 (3.10 m)	12 times	14 seconds	
Motion 3 (2.52 m)	10 times	14 seconds	
Motion 4 (3.10 m)	12 times	13 seconds	
Total	47 times	57 seconds	
6^{th} Data			
Motion 1 (2.52 m)	14 times	16 seconds	800 ms
Motion 2 (3.10 m)	13 times	14 seconds	
Motion 3 (2.52 m)	12 times	15 seconds	
Motion 4 (3.10 m)	13 times	15 seconds	
Total	52 times	60 seconds	
7^{th} Data			
Motion 1 (2.52 m)	14 times	15 seconds	800 ms
Motion 2 (3.10 m)	12 times	14 seconds	
Motion 3 (2.52 m)	11 times	14 seconds	
Motion 4 (3.10 m)	12 times	13 seconds	
Total	49 times	56 seconds	
8^{th} Data			
Motion 1 (2.52 m)	13 times	14 seconds	800 ms
Motion 2 (3.10 m)	12 times	13 seconds	
Motion 3 (2.52 m)	11 times	14 seconds	
Motion 4 (3.10 m)	12 times	14 seconds	
Total	48 times	55 seconds	
9^{th} Data			
Motion 1 (2.52 m)	14 times	16 seconds	800 ms
Motion 2 (3.10 m)	12 times	14 seconds	
Motion 3 (2.52 m)	10 times	13 seconds	
Motion 4 (3.10 m)	13 times	14 seconds	
Total	49 times	57 seconds	

10^{th} Data			
Motion 1 (2.52 m)	14 times	16 seconds	800 ms
Motion 2 (3.10 m)	13 times	15 seconds	
Motion 3 (2.52 m)	12 times	14 seconds	
Motion 4 (3.10 m)	12 times	15 seconds	
Total	51 times	60 seconds	

4.3.3 Room C (L-Shape Room)

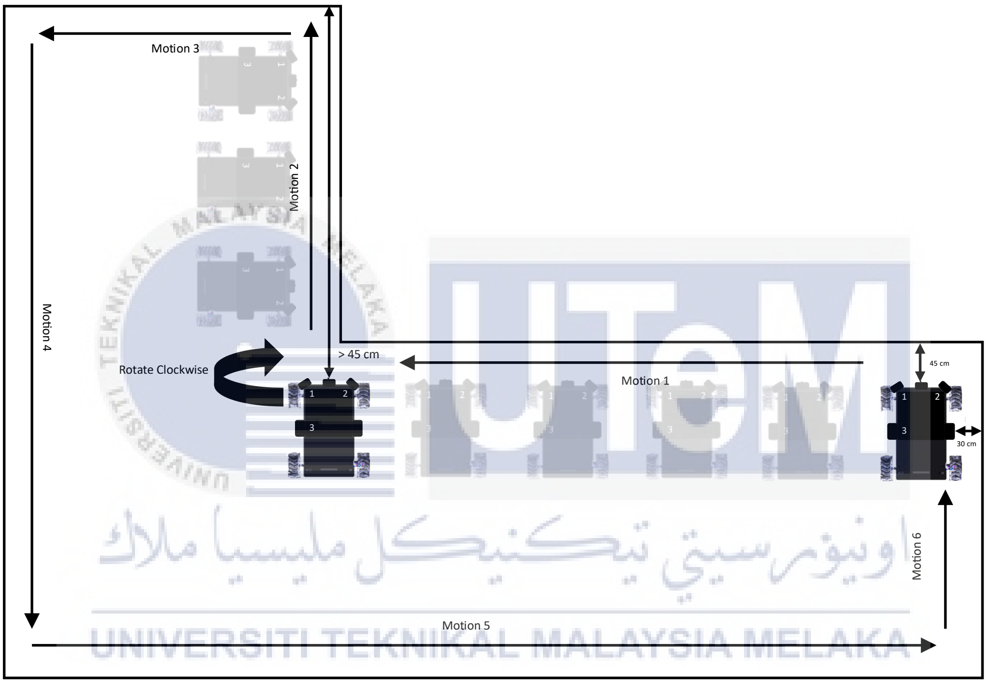


Figure 4.6: Mobile Robot Slide Left in L-Shape Room

Dimensions of room C,

Length x Width = 3.30-meter x 8.15-meters

L-shape dimensions Length = 1.67-meter (Motion 1) x 1.61-meters (Motion 2) x 1.63-meters (Motion 3) x 8.15-meters (Motion 4) x 3.30-meters (Motion 5) x 6.54-meters (Motion 6)

Time of Flight Sensor detects distance = 45 cm

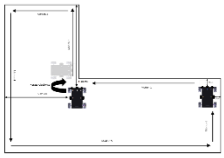
Ultrasonic Sensor detects distance = 45 cm

Distance length robot slide left for LENGTH L-SHAPE, 1.67 m – 0.3 m = 1.37 m

Distance length robot slide left for WIDTH L-SHAPE, 1.61 m – 0.45 m = 1.16 m

Distance length robot slide left for MOTION 3, 1.63 m – (0.45 + 0.45) m = 0.73 m

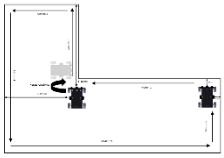
Table 4.5: Data Calculation Movement Robot for L-Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
1^{st} Data			
Motion 1 (1.67 m)	14 times	14 seconds	
Motion 2 (1.61 m)	10 times	8 seconds	
Motion 3 (1.63 m)	11 times	10 seconds	
Motion 4 (8.15 m)	86 times	48 seconds	400 ms
Motion 5 (3.30 m)	27 times	24 seconds	
Motion 6 (6.54 m)	71 times	41 seconds	
Total	219 times	145 seconds	
2^{nd} Data			
Motion 1 (1.67 m)	15 times	16 seconds	
Motion 2 (1.61 m)	11 times	9 seconds	
Motion 3 (1.63 m)	12 times	11 seconds	
Motion 4 (8.15 m)	82 times	45 seconds	400 ms
Motion 5 (3.30 m)	30 times	28 seconds	
Motion 6 (6.54 m)	75 times	43 seconds	
Total	225 times	152 seconds	
3^{rd} Data			
Motion 1 (1.67 m)	13 times	13 seconds	
Motion 2 (1.61 m)	9 times	7 seconds	
Motion 3 (1.63 m)	10 times	9 seconds	
Motion 4 (8.15 m)	88 times	50 seconds	400 ms
Motion 5 (3.30 m)	25 times	22 seconds	
Motion 6 (6.54 m)	68 times	39 seconds	
Total	223 times	140 seconds	

4^{th} Data			
Motion 1 (1.67 m)	16 times	15 seconds	400 ms
Motion 2 (1.61 m)	10 times	8 seconds	
Motion 3 (1.63 m)	11 times	10 seconds	
Motion 4 (8.15 m)	80 times	47 seconds	
Motion 5 (3.30 m)	28 times	25 seconds	
Motion 6 (6.54 m)	73 times	42 seconds	
Total	218 times	147 seconds	
5^{th} Data			
Motion 1 (1.67 m)	14 times	12 seconds	400 ms
Motion 2 (1.61 m)	12 times	9 seconds	
Motion 3 (1.63 m)	13 times	11 seconds	
Motion 4 (8.15 m)	85 times	49 seconds	
Motion 5 (3.30 m)	29 times	26 seconds	
Motion 6 (6.54 m)	70 times	40seconds	
Total	223 times	147 seconds	
6^{th} Data			
Motion 1 (1.67 m)	15 times	14 seconds	400 ms
Motion 2 (1.61 m)	11 times	8 seconds	
Motion 3 (1.63 m)	10 times	9 seconds	
Motion 4 (8.15 m)	83 times	46 seconds	
Motion 5 (3.30 m)	31 times	28 seconds	
Motion 6 (6.54 m)	74 times	41 seconds	
Total	224 times	146 seconds	
7^{th} Data			
Motion 1 (1.67 m)	13 times	13 seconds	400 ms
Motion 2 (1.61 m)	9 times	7 seconds	
Motion 3 (1.63 m)	12 times	11 seconds	
Motion 4 (8.15 m)	87 times	48 seconds	
Motion 5 (3.30 m)	27 times	24 seconds	
Motion 6 (6.54 m)	69 times	40 seconds	
Total	217 times	143 seconds	

8th Data			
Motion 1 (1.67 m)	16 times	15 seconds	400 ms
Motion 2 (1.61 m)	10 times	8 seconds	
Motion 3 (1.63 m)	11 times	10 seconds	
Motion 4 (8.15 m)	81 times	46 seconds	
Motion 5 (3.30 m)	29 times	26 seconds	
Motion 6 (6.54 m)	72 times	41 seconds	
Total	219 times	146 seconds	
9th Data			
Motion 1 (1.67 m)	14 times	12 seconds	400 ms
Motion 2 (1.61 m)	12 times	9 seconds	
Motion 3 (1.63 m)	13 times	11 seconds	
Motion 4 (8.15 m)	84 times	47 seconds	
Motion 5 (3.30 m)	28 times	25 seconds	
Motion 6 (6.54 m)	71 times	40 seconds	
Total	222 times	144 seconds	
10th Data			
Motion 1 (1.67 m)	15 times	14 seconds	400 ms
Motion 2 (1.61 m)	11 times	8 seconds	
Motion 3 (1.63 m)	10 times	9 seconds	
Motion 4 (8.15 m)	82 times	45 seconds	
Motion 5 (3.30 m)	30 times	28 seconds	
Motion 6 (6.54 m)	73 times	42 seconds	
Total	221 times	146 seconds	

Table 4.6: Data Calculation Movement Robot for L-Shape Room

	**Total of Slide Lefts**	**Time to Complete One Cycle (s)**	**Sliding delay time (ms)**
1st Data			
Motion 1 (1.67 m)	12 times	12 seconds	
Motion 2 (1.61 m)	8 times	7 seconds	
Motion 3 (1.63 m)	9 times	8 seconds	
Motion 4 (8.15 m)	72 times	40 seconds	800 ms
Motion 5 (3.30 m)	23 times	21 seconds	
Motion 6 (6.54 m)	60 times	34 seconds	
Total	184 times	122 seconds	
2nd Data			
Motion 1 (1.67 m)	13 times	14 seconds	
Motion 2 (1.61 m)	9 times	8 seconds	
Motion 3 (1.63 m)	10 times	9 seconds	
Motion 4 (8.15 m)	70 times	38 seconds	800 ms
Motion 5 (3.30 m)	25 times	23 seconds	
Motion 6 (6.54 m)	65 times	37 seconds	
Total	192 times	129 seconds	
3rd Data			
Motion 1 (1.67 m)	11 times	11 seconds	
Motion 2 (1.61 m)	7 times	6 seconds	
Motion 3 (1.63 m)	8 times	7 seconds	
Motion 4 (8.15 m)	74 times	42 seconds	800 ms
Motion 5 (3.30 m)	21 times	19 seconds	
Motion 6 (6.54 m)	56 times	32 seconds	
Total	177 times	117 seconds	

4th Data			
Motion 1 (1.67 m)	14 times	13 seconds	
Motion 2 (1.61 m)	8 times	7 seconds	
Motion 3 (1.63 m)	9 times	8 seconds	
Motion 4 (8.15 m)	68 times	39 seconds	800 ms
Motion 5 (3.30 m)	24 times	22 seconds	
Motion 6 (6.54 m)	62 times	35 seconds	
Total	185 times	124 seconds	
5th Data			
Motion 1 (1.67 m)	12 times	10 seconds	
Motion 2 (1.61 m)	10 times	8 seconds	
Motion 3 (1.63 m)	11 times	10 seconds	
Motion 4 (8.15 m)	75 times	43 seconds	800 ms
Motion 5 (3.30 m)	26 times	24 seconds	
Motion 6 (6.54 m)	63 times	36 seconds	
Total	197 times	121 seconds	
6th Data			
Motion 1 (1.67 m)	13 times	12 seconds	
Motion 2 (1.61 m)	9 times	8 seconds	
Motion 3 (1.63 m)	8 times	7 seconds	
Motion 4 (8.15 m)	71 times	41 seconds	800 ms
Motion 5 (3.30 m)	28 times	26 seconds	
Motion 6 (6.54 m)	67 times	38 seconds	
Total	196 times	132 seconds	
7th Data			
Motion 1 (1.67 m)	11 times	11 seconds	
Motion 2 (1.61 m)	7 times	6 seconds	
Motion 3 (1.63 m)	10 times	9 seconds	
Motion 4 (8.15 m)	73 times	42 seconds	800 ms
Motion 5 (3.30 m)	23 times	21 seconds	
Motion 6 (6.54 m)	58 times	33 seconds	
Total	182 times	122 seconds	

8th Data			
Motion 1 (1.67 m)	14 times	13 seconds	800 ms
Motion 2 (1.61 m)	8 times	7 seconds	
Motion 3 (1.63 m)	9 times	8 seconds	
Motion 4 (8.15 m)	69 times	40 seconds	
Motion 5 (3.30 m)	25 times	23 seconds	
Motion 6 (6.54 m)	61 times	35 seconds	
Total	186 times	126 seconds	
9th Data			
Motion 1 (1.67 m)	12 times	10 seconds	800 ms
Motion 2 (1.61 m)	10 times	8 seconds	
Motion 3 (1.63 m)	11 times	10 seconds	
Motion 4 (8.15 m)	74 times	43 seconds	
Motion 5 (3.30 m)	24 times	22 seconds	
Motion 6 (6.54 m)	60 times	34 seconds	
Total	191 times	127 seconds	
10th Data			
Motion 1 (1.67 m)	13 times	12 seconds	800 ms
Motion 2 (1.61 m)	9 times	8 seconds	
Motion 3 (1.63 m)	8 times	7 seconds	
Motion 4 (8.15 m)	70 times	38 seconds	
Motion 5 (3.30 m)	26 times	24 seconds	
Motion 6 (6.54 m)	63 times	36 seconds	
Total	189 times	125 seconds	

For this part, there are two tables (400 ms speed & 800 ms speed) above showing Data Calculation Movement Robot. there are four motions for Rectangle Room and six motions for L-Shape Room. Each dimension needs to record data for the robot to complete one motion in second with the total of the slide left movement robot. And also, this calculation to make the real tasks for the robot complete the movement with the total slide left and time robot to complete in every motion.

4.3.4 Graph Experiment Data Movement Mobile Robot

This is the result of the graph obtained from the experimental data. The data obtained is from the result of the movement of the speed of the motor completing a round in one room.

4.3.4.1 Analysis Graph of Room A, Room B and Room C

Graph Analysis Room A, Room B and Room C are the results of the data obtained. Based on the graph below showing total data from Total of Slide Lefts and Time to Complete One Cycle.

i. Room A (Rectangle Shape Room)

Length x Width = 3.18-meter x 3.20-meters

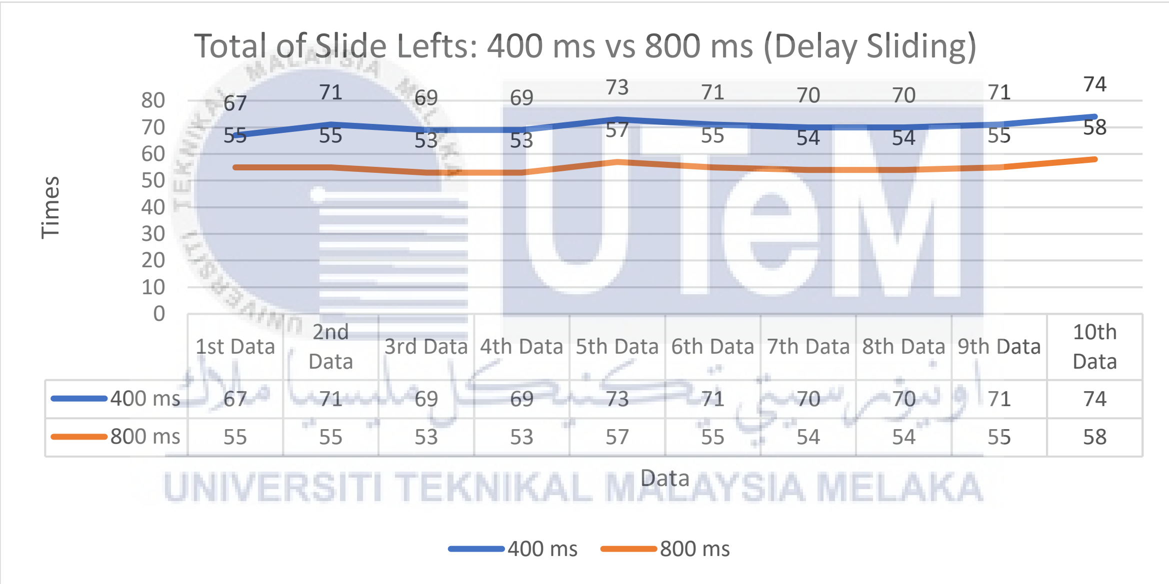


Figure 4.7:Total of Slide Lefts at Rectangle Shape Room A

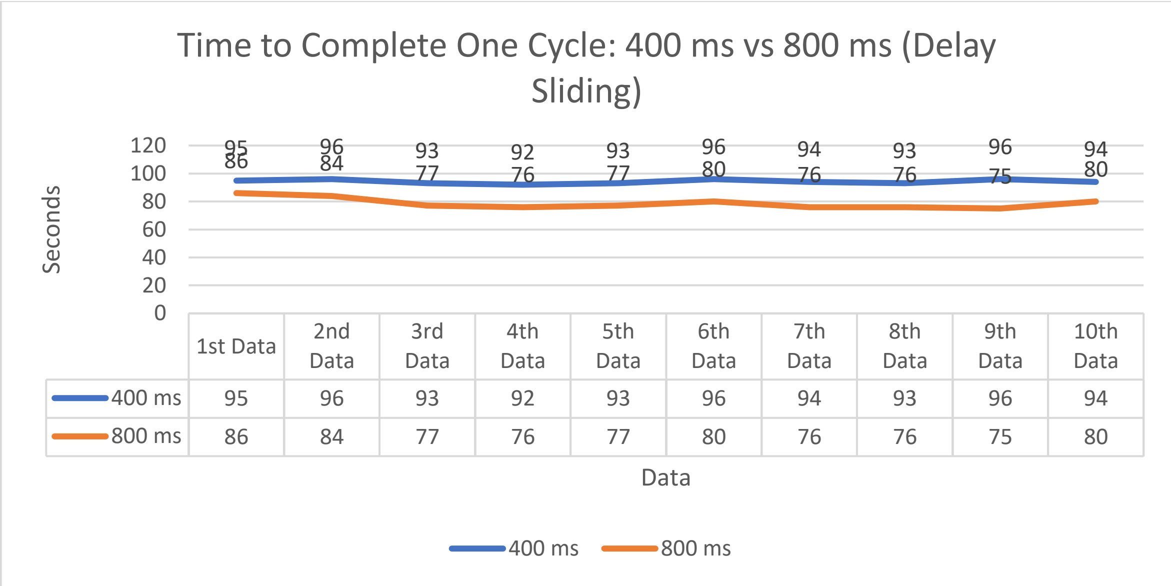


Figure 4.8: Time to Complete One Cycle at Rectangle Shape Room A

ii. Room B (Rectangle Shape Room)

Length x Width = 2.52-meter x 3.10-meters

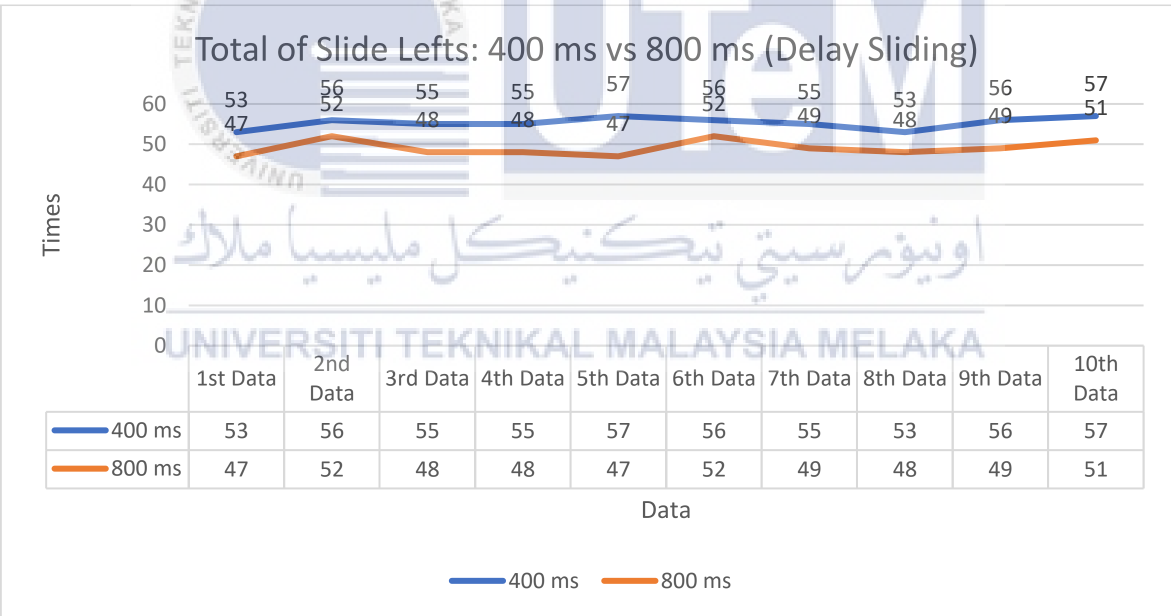


Figure 4.9: Total of Slide Lefts at Rectangle Shape Room B

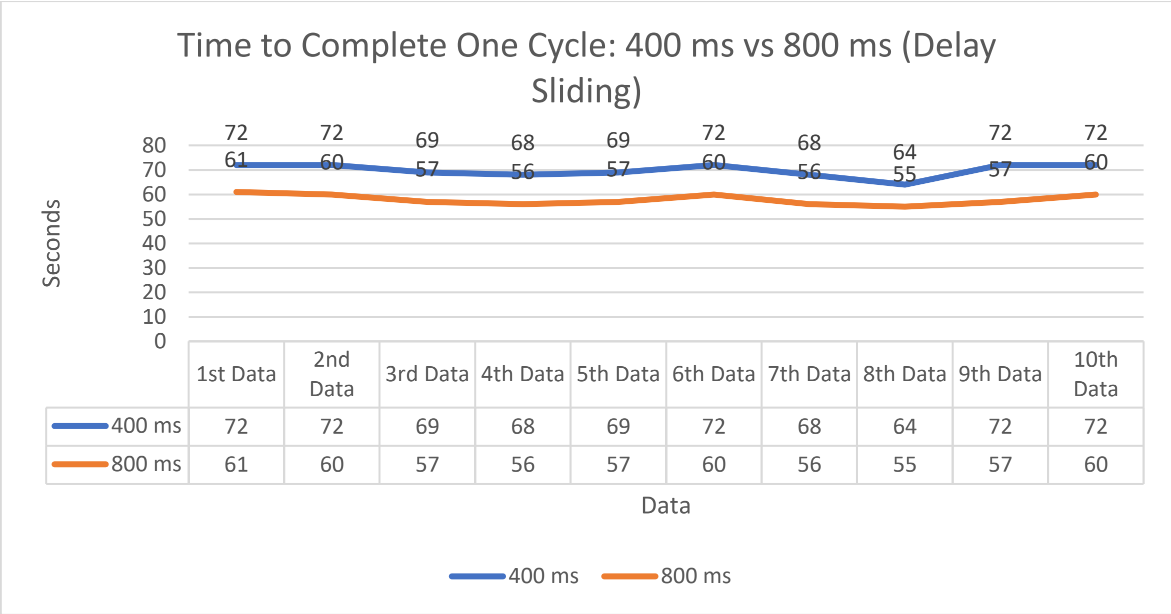


Figure 4.10: Time to Complete One Cycle at Rectangle Shape Room B

iii. Room C (L Shape Room)

Length x Width = 3.30-meter x 8.15-meters

L-shape dimensions Length = 1.67-meter (Motion 1) x 1.61-meters (Motion 2) x 1.63-meters

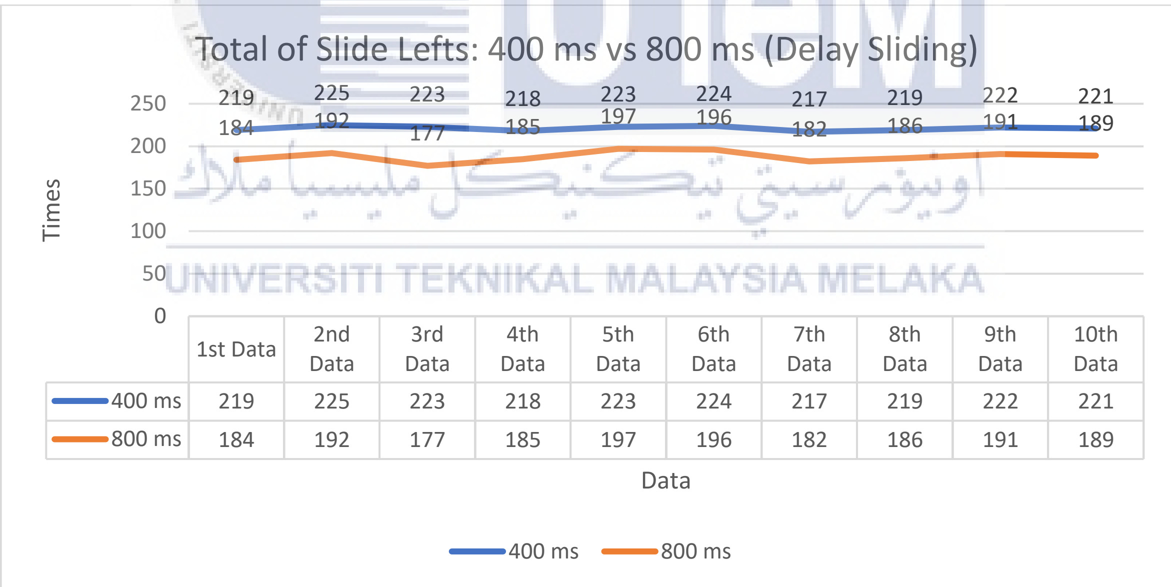


Figure 4.11: Total of Slide Lefts at L-Shape Room C

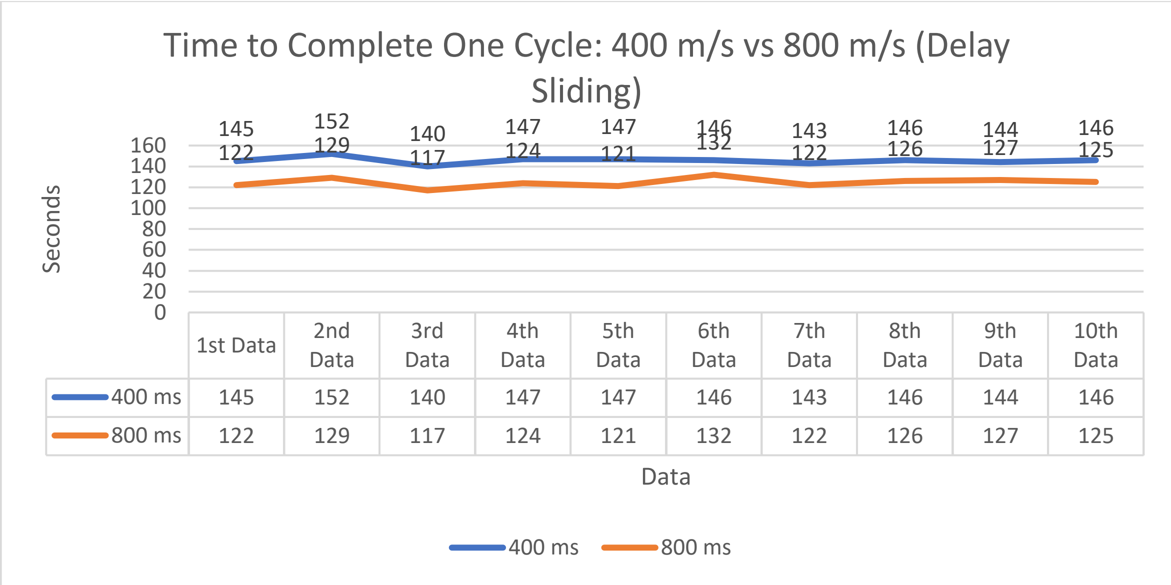


Figure 4.12: Time to Complete One Cycle at L-Shape Room C

From the analysis graph from Figure 4.7 to Figure 4.12, the data collected from an experiment conducted in Room A, a rectangular-shaped room with dimensions 3.18 meters by 3.20 meters, Room B, a rectangular-shaped room with dimensions 2.52 meters by 3.10 meters and Room C, a L-shaped room with dimensions 3.30 meters by 8.15 meters but for L-shaped room the length from Motion 1, Motion 2 and Motion 3 with dimensions 1.62 meters, 1.61 meters and 1.63 meters. The experiment involves measuring the total slide left times at two different delay sliding, 400 ms and 800 ms. The x-axis of the graph both graphs (Total of Slide Lefts and Time to Complete One Cycle) procced the ten sets of data, while the y-axis quantifies the times and seconds recorded during each set of data collection. Briefly, it is evident that slide left times and times to complete one cycle are consistently lower at a delay sliding of 800 ms compared to those recorded at 400 ms. For instance, during the first set of data collection of all graphs, it shows a difference in value from 400 ms to 800 ms, to complete slide lefts and one cycle of rooms. This value continues throughout all ten sets of data, every time, fewer counts are recorded at the higher delay. This consistent graph shows that increasing delay significantly reduces the time required to complete the left slide and the time to complete one cycle in Room A, Room B and Room C. It can be concluded that this graph will continue with increasing delay. In conclusion, based on this specific data set and under these specific conditions (in the dimensions for the three types of rooms), increasing the speed from 400 ms to 800 ms results in a significant reduction.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, based on the three objectives that have been completed according to the objective, designing the robot, development algorithms for the robot to follow the wall, and evaluating the performance robot. The experiment results show the robot successfully moved in the two types of rooms, namely Rectangle Shape Room and L-Shape Room. The use of ultrasonic sensors and Time-of-Flight sensors proved effective in detecting and measuring distances between robots and walls. The theoretical calculations of the robot's movement were compared with the experiment data. Then, the experiment showed that the sliding motion time significantly reduced the time for complete total sliding the motion each wall. The findings show that the robots in this project have the potential to perform the building quality assessment process, providing faster and more consistent assessments compared to human assessors. The integration of sensors and actuators in the design of the robot, along with navigation in different room shapes, showcases the robot's adaptability and efficiency.

5.2 Future Works

To improve the functionality and performance of mobile robot navigation for building quality assessment, future work should focus on several key areas. First, it's useful to explore and fully utilize the Arduino UNO platform to control robot movements. For that purpose, it is necessary to use the functions of Arduino UNO and develop and specify an algorithm that can accurately and reliably control the movement of the robot. Additionally, it is important to identify and integrate the right components that can effectively perform building inspections. This may include research and selection of sensors, cameras and other technologies that can provide accurate information about build quality. Additionally, investing time in learning Visual Studio and building your application can greatly improve the user interface and interaction with your mobile robot navigation system. The application offers intuitive controls, real-time data visualization, and comprehensive reporting capabilities, enabling users to effectively monitor and assess building quality. Extensive testing should be performed to improve the overall performance and smooth movement of mobile robots. This requires improved navigation algorithms, optimized path planning, and the ability for robots to effectively follow walls and designated routes. Through rigorous testing and performance evaluation, deficiencies and limitations in robot movement can be identified and corrected, resulting in a more reliable and efficient system. In summary, this future focus can further improve mobile robotic navigation systems for construction quality assessment, bringing further accuracy, productivity and added value to the construction industry.

REFERENCES

[1] *Local logic optimization algorithm for autonomous mobile robot based on fuzzy logic*. (2018, June 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8407847

[2] Salih, J. E. M., Rizon, M., Yaacob, S., Adom, A. H., & Mamat, M. N. (2006). Designing Omni-Directional Mobile Robot with Mecanum Wheel. *American Journal of Applied Sciences*, *3*(5), 1831–1835. https://doi.org/10.3844/ajassp.2006.1831.1835

[3] Taheri, H. R., Qiao, B., & Ghaeminezhad, N. (2015). Kinematic Model of a Four Mecanum Wheeled Mobile Robot. *International Journal of Computer Applications*, *113*(3), 6–9. https://doi.org/10.5120/19804-1586

[4] *Four Wheeled Mobile Robots: A Review*. (2022, September 13). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9936855

[5] Santos, A.G. (2008). Autonomous Mobile Robot Navigation using Smartphones.

[6] Nugroho, C. Y., Nugraha, R., & Rusdinar, A. (2017). *Robot control design using virtual path from android smartphone*. https://doi.org/10.1109/icfst.2017.8210550

[7] Hazry, D., Sugisaka, M., & Yuji, T. (2006). Human-Robot Interface over the Web Based Intelligent System. *American Journal of Applied Sciences*, *3*(1), 1634–1639. https://doi.org/10.3844/ajassp.2006.1634.1639

[8] Vanitha, M., M, S., & Selvarasu, R. (2016). *Monitoring and controlling of mobile robot via internet through raspberry Pi board*. https://doi.org/10.1109/iconstem.2016.7560864

[9] Banduka, M. L. (2015). Remote Monitoring and Control of Industrial Robot based on Android Device and Wi-Fi Communication. *Automatika*, *56*(3). https://doi.org/10.7305/automatika.2015.10.1057

[10] Fahmidur, R. K., Munaim, H. M. A., Tanvir, S. M., & Sayem, A. S. (2016). *Internet controlled robot: A simple approach*. https://doi.org/10.1109/iceeot.2016.7754873

[11] Rasha, R., Mostafa, A., & Motahar, T. (2018). Secret Bot: A Bluetooth Controlled Spy Robot. *ResearchGate*. https://www.researchgate.net/publication/337672789_Secret_Bot_A_Bluetooth_Controlled_Spy_Robot

[12] UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices. (n.d.). *www.analog.com*. https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html#:~:text=By%20definition%2C%20UART%20is%20a,going%20to%20the%20receiving%20end.

[13] *Bluetooth based home automation system using cell phone*. (2011, June 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5973811

[14] Handayani, Y. S., & Mardiana, Y. (2018). KENDALI ROBOT BLUETOOTH DENGAN SMARTPHONE ANDROID BERBASIS ARDUINO UNO. *Ilkom Jurnal Ilmiah*. https://doi.org/10.33096/ilkom.v10i3.363.331-337

[15] *Raspberry Pi Introduction | Raspberry Pi*. (n.d.). © 2018 ElectronicWings. https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction

[16] Instructables. (2020). Intro to Arduino. *Instructables*. https://www.instructables.com/Intro-to-Arduino/

[17] *introduction to sensors - Google Search*. (n.d.). https://www.google.com/search?q=introduction+to+sensors&oq=Introduction+to+sensors&aqs=chrome.0.0i512j0i20i263i512j0i512l2j46i512j0i512j69i60l2.490j0j7&sourceid=chrome&ie=UTF-8

[18] Ng, F. (2020). Ultrasonic Sensors. *ResearchGate*. https://doi.org/10.13140/RG.2.2.33638.78404

[19] Peng, M. (n.d.). *RPlidar A1 – SLAMTEC Global Network*. https://www.slamtec.com/en/Lidar/A1

[20] *RPLIDAR A1 Low Cost 360 Degree Laser Range Scanner Development Kit User Manual - Google Search*. (n.d.). https://www.google.com/search?q=RPLIDAR+A1+Low+Cost+360+Degree+Laser+Range+Scanner+Development+Kit+User+Manual&oq=RPLIDAR+A1+Low+Cost+360+Degree+Laser+Range+Scanner+Development+Kit+User+Manual&aqs=chrome.69i57.419j0j7&sourceid=chrome&ie=UTF-8

[21] Alenzi, Z., Alenzi, E., Alqasir, M., Alruwaili, M., Alhmiedat, T., & Alia, O. M. (2022). A Semantic Classification Approach for Indoor Robot Navigation. *Electronics*, *11*(13), 2063. https://doi.org/10.3390/electronics11132063

[22] *Infrared Sensor*. (n.d.). https://www.infratec.eu/sensor-division/service-support/glossary/infrared-sensor/#:~:text=An%20infrared%20sensor%20(IR%20sensor,systems%20to%20detect%20unwelcome%20guests.

[23] *Microsoft Kinect Sensor and Its Effect*. (2012, February 1). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/6190806

[24] Ballester, J. C., & Pheatt, C. (2013). Using the Xbox Kinect sensor for positional data acquisition. *American Journal of Physics*, *81*(1), 71–77. https://doi.org/10.1119/1.4748853

[25] Leal-Meléndrez, J. A., Altamirano-Robles, L., & Gonzalez, J. A. (2013). Occlusion Handling in Video-Based Augmented Reality Using the Kinect Sensor for Indoor Registration. In *Lecture Notes in Computer Science* (pp. 447–454). Springer Science+Business Media. https://doi.org/10.1007/978-3-642-41827-3_56

[26] *Bluetooth low energy profile for MPU9150 IMU data transfers*. (2017). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/7889431

[27] *Gesture detection and data fusion based on MPU9250 sensor*. (2015, July 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/7494488

[28] Okubo, Y., Ye, C., & Borenstein, J. (2009). Characterization of the Hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation. In *Proceedings of SPIE*. SPIE. https://doi.org/10.1117/12.818332

[29] *Characterization of the compact Hokuyo URG-04LX 2D laser range scanner*. (2009, May 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5152579

[30] *Characterization of the Hokuyo UBG-04LX-F01 2D laser rangefinder*. (2010, September 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5598672

[31] Passeraub, P. (1999). An integrated inductive proximity sensor. *An Integrated Inductive Proximity Sensor*. https://doi.org/10.5075/epfl-thesis-1939

[32] *Extending the Operating Distance of Inductive Proximity Sensor Using Magnetoplated Wire*. (2009, October 1). https://ieeexplore.ieee.org/abstract/document/5257099

[33] *ultrasonic sensor datasheet pdf - Google Search*. (n.d.). https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[34] *RPLIDAR A 1 Low Cost 360 Degree Laser Range Scanner Development Kit User Manual - Google Search*. (n.d.). https://bucket-download.slamtec.com/e680b4e2d99c4349c019553820904f28c7e6ec32/LM108_SLAMTEC_rplidarkit_usermaunal_A1M8_v1.0_en.pdf

[35] *ir proximity sensor datasheet - Google Search*. (n.d.). https://components101.com/sites/default/files/component_datasheet/Datasheet%20of%20IR%20%20Sensor.pdf

[36] Davudinasab, A. (2014). Kinect Sensor. *ResearchGate*. https://doi.org/10.13140/2.1.1068.5124

[37] *Scanning Laser Range Finder URG-04LX Specifications - Google Search*. (n.d.). https://www.hokuyo-aut.jp/dl/Specifications_URG-04LX_1513063395.pdf

[38] *inductive proximity sensor datasheet pdf - Google Search*. (n.d.). https://docs.rs-online.com/e8de/0900766b80274a6e.pdf

[39] Yatmono, S., & Muhfizaturrahmah. (2017b). *Developing Design of Mobile Robot Navigation System based on Android*. https://doi.org/10.2991/ictvt-17.2017.19

[40] Alenzi, Z., Alenzi, E., Alqasir, M., Alruwaili, M., Alhmiedat, T., & Alia, O. M. (2022b). A Semantic Classification Approach for Indoor Robot Navigation. *Electronics*, *11*(13), 2063. https://doi.org/10.3390/electronics11132063

[41] Oltean, S. (2019b). Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation. *Procedia Manufacturing*, *32*, 572–577. https://doi.org/10.1016/j.promfg.2019.02.254

[42] Pandey, A., Kumar, S., Pandey, K. M., & Parhi, D. R. (2016). Mobile robot navigation in unknown static environments using ANFIS controller. *Perspectives in Science*, *8*, 421–423. https://doi.org/10.1016/j.pisc.2016.04.094

[43] Azeta, J., Bolu, C., Hinvi, D., & Abioye, A. A. (2019b). Obstacle detection using ultrasonic sensor for a mobile robot. *IOP Conference Series: Materials Science and Engineering*, *707*(1), 012012. https://doi.org/10.1088/1757-899x/707/1/012012

[44] Abueejela, Y. M., & Ali, H. T. M. (2018d). Wheeled Mobile Robot Obstacle Avoidance Using Compass and Ultrasonic. *Universal Journal of Control and Automation*, *6*(1), 13–18. https://doi.org/10.13189/ujca.2018.060102

[45] Mishra, D. K., Thomas, A. K., Kuruvilla, J., Kalyanasundaram, P., Prasad, K. R., & Haldorai, A. (2022b). Design of mobile robot navigation controller using neuro-fuzzy logic system. *Computers & Electrical Engineering*, *101*, 108044. https://doi.org/10.1016/j.compeleceng.2022.108044

[46] Petavratzis, E., Volos, C., & Stouboulos, I. (2023b). Experimental study of terrain coverage of an autonomous chaotic mobile robot. *Integration*, *90*, 104–114. https://doi.org/10.1016/j.vlsi.2023.01.010

[47] Patle, B., Sanap, S., Jha, A., Patel, B., & Kashyap, S. K. (2021b). Metric-Cauchy sequence for wheeled mobile robot navigation. *Materials Today: Proceedings*, *38*, 3213–3219. https://doi.org/10.1016/j.matpr.2020.09.660

[48] Manikandan, N. S., & Kaliyaperumal, G. (2022). Collision avoidance approaches for autonomous mobile robots to tackle the problem of pedestrians roaming on campus

road. *Pattern Recognition Letters*, *160*, 112–121. https://doi.org/10.1016/j.patrec.2022.06.005

[49] Sankar, J., Adarsh, S., & Ramachandran, K. (2018b). Performance Evaluation of Ultrasonic and Infrared Waves on Human Body and Metal Surfaces for Mobile Robot Navigation. *Materials Today: Proceedings*, *5*(8), 16516–16525. https://doi.org/10.1016/j.matpr.2018.06.007

[50] *An Inexpensive Autonomous Mobile Robot for Undergraduate Education: Integration of Arduino and Hokuyo Laser Range Finders*. (2022). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/9841571

[51] Khairudin, M., Sp, H., Karim, H. I., & Azman, M. N. A. (2020b). Vision-Based Mobile Robot Navigation For Suspicious Object Monitoring In Unknown Environments. *ResearchGate*. https://www.researchgate.net/publication/338843066_VISION-BASED_MOBILE_ROBOT_NAVIGATION_FOR_SUSPICIOUS_OBJECT_MONITORING_IN_UNKNOWN_ENVIRONMENTS

[52] *Embedded Design and Implementation of Mobile Robot for Surveillance Applications - Google Search*. (n.d.). https://seap.taylors.edu.my/file/rems/publication/100359_8344_1.pdf

[53] Azeta, J., Bolu, C., Hinvi, D., Abioye, A. A., Boyo, H., Anakhu, P. I., & Onwordi, P. (2019). An Android Based Mobile Robot for Monitoring and Surveillance. *Procedia Manufacturing*, *35*, 1129–1134. https://doi.org/10.1016/j.promfg.2019.06.066

[54] *Robotic snake | 3D CAD Model Library | GrabCAD*. (n.d.). https://grabcad.com/library/robotic-snake-1

[55] *Robot dog | 3D CAD Model Library | GrabCAD*. (n.d.). https://grabcad.com/library/robot-dog-10

[56] *ROBOT CAR | 3D CAD Model Library | GrabCAD*. (n.d.). https://grabcad.com/library/robot-car-1

APPENDICES

APPENDIX 1 ARDUINO CODE

a. Arduino Code Rectangle Shape Room

```
#include "Adafruit_VL53L0X.h"
// address we will assign if dual sensor is present
#define LOX1_ADDRESS 0x30
#define LOX2_ADDRESS 0x31
#include <HCSR04.h>

HCSR04 hc(A2, A3);

// set the pins to shutdown
#define SHT_LOX1 A0
#define SHT_LOX2 A1

// objects for the vl53l0x
Adafruit_VL53L0X lox1 = Adafruit_VL53L0X();
Adafruit_VL53L0X lox2 = Adafruit_VL53L0X();

// this holds the measurement
VL53L0X_RangingMeasurementData_t measure1;
VL53L0X_RangingMeasurementData_t measure2;

#include <AFMotor.h>
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

int jarak;
int offset;
int sen1, sen2;

void setID() {
  // all reset
  digitalWrite(SHT_LOX1, LOW);
  digitalWrite(SHT_LOX2, LOW);
//  digitalWrite(SHT_LOX3, LOW);
  delay(10);
  // all unreset
  digitalWrite(SHT_LOX1, HIGH);
  digitalWrite(SHT_LOX2, HIGH);
  //digitalWrite(SHT_LOX3, HIGH);
  delay(10);

  // activating LOX1 and reseting LOX2
```

```
  digitalWrite(SHT_LOX1, HIGH);
  digitalWrite(SHT_LOX2, LOW);
  //digitalWrite(SHT_LOX3, LOW);

  // initing LOX1
  if(!lox1.begin(LOX1_ADDRESS)) {
    Serial.println(F("Failed to boot first VL53L0X"));
    while(1);
  }
  delay(10);

  // activating LOX2
  digitalWrite(SHT_LOX2, HIGH);
  delay(10);

  //initing LOX2
  if(!lox2.begin(LOX2_ADDRESS)) {
    Serial.println(F("Failed to boot second VL53L0X"));
    while(1);
  }

  lox1.configSensor(Adafruit_VL53L0X::VL53L0X_SENSE_LONG_RANGE);
  lox1.configSensor(Adafruit_VL53L0X::VL53L0X_SENSE_LONG_RANGE);

}

void read2sensors() {

  lox1.rangingTest(&measure1, false); // pass in 'true' to get debug data printout!
  lox2.rangingTest(&measure2, false); // pass in 'true' to get debug data printout!

  // print sensor one reading
  Serial.print(F("1: "));
  if(measure1.RangeStatus != 4) {     // if not out of range
   sen1 = measure1.RangeMilliMeter;
    Serial.print(sen1);
  } else {    Serial.print(F("999"));  }
 Serial.print(F(" "));

  // print sensor two reading
  Serial.print(F("2: "));
  if(measure2.RangeStatus != 4) {
    sen2 = measure2.RangeMilliMeter;
    Serial.print(sen2);
  } else {    Serial.print(F("999"));  }
  Serial.println();
}

void robotstop()
{
```

```
 motor1.run(FORWARD);   motor1.setSpeed(0);  motor2.run(FORWARD);
motor2.setSpeed(0);
 motor3.run(FORWARD);   motor3.setSpeed(0);  motor4.run(FORWARD);
motor4.setSpeed(0);
}

void robotfwd(int speed)
{
 motor1.run(FORWARD);   motor1.setSpeed(speed);  motor2.run(FORWARD);
motor2.setSpeed(speed);
 motor3.run(FORWARD);   motor3.setSpeed(speed+40);  motor4.run(FORWARD);
motor4.setSpeed(speed+20);
}

void robotrev(int speed)
{
 motor1.run(BACKWARD);   motor1.setSpeed(speed);  motor2.run(BACKWARD);
motor2.setSpeed(speed);
 motor3.run(BACKWARD);   motor3.setSpeed(speed+40);
motor4.run(BACKWARD);   motor4.setSpeed(speed+20);
}

void robotright(int speed)
{
 motor1.run(FORWARD);   motor1.setSpeed(speed);  motor2.run(BACKWARD);
motor2.setSpeed(speed);
 motor3.run(BACKWARD);   motor3.setSpeed(speed+40);  motor4.run(FORWARD);
motor4.setSpeed(speed+20);
}

void robotleft(int speed)
{
 motor1.run(BACKWARD);   motor1.setSpeed(speed);  motor2.run(FORWARD);
motor2.setSpeed(speed);
 motor3.run(FORWARD);   motor3.setSpeed(speed+40);  motor4.run(BACKWARD);
motor4.setSpeed(speed+20);
}

void robotslideright(int speed)
{
 motor1.run(FORWARD);   motor1.setSpeed(speed);  motor2.run(BACKWARD);
motor2.setSpeed(speed);
 motor3.run(FORWARD);   motor3.setSpeed(speed+40);  motor4.run(BACKWARD);
motor4.setSpeed(speed+20);
}

void robotslideleft(int speed)
{
 motor1.run(BACKWARD);   motor1.setSpeed(speed);  motor2.run(FORWARD);
motor2.setSpeed(speed);
```

```
 motor3.run(BACKWARD);    motor3.setSpeed(speed+40);  motor4.run(FORWARD);
motor4.setSpeed(speed+20);
}

void setup() {
  jarak = 450;
  offset = 50;
  Serial.begin(115200);
  // wait until serial port opens for native USB devices
  while (! Serial) { delay(1); }
  pinMode(SHT_LOX1, OUTPUT);  pinMode(SHT_LOX2, OUTPUT);
//pinMode(SHT_LOX3, OUTPUT);
  Serial.println(F("Shutdown pins inited..."));
  digitalWrite(SHT_LOX1, LOW);  digitalWrite(SHT_LOX2, LOW);
//digitalWrite(SHT_LOX3, LOW);
  Serial.println(F("Both in reset mode...(pins are low)"));
  Serial.println(F("Starting..."));
  setID();

}


void alignfront(int dist)
{
  int cnt=0; int diff1=0; int diff2=0;
  read2sensors();
 while(cnt==0)
  {
    read2sensors(); diff1 = sen2 - sen1;
    if(sen1 > dist && sen2 > dist)    {robotfwd(70);     }
    if (sen1 < dist && sen2 <dist)
    {
      if(sen1 < dist-100 && sen2 < dist-100)    {robotrev(70);      }
    else
      {robotstop(); cnt =1;
      while(cnt == 1)
      {
       read2sensors(); diff1 = sen2 - sen1;
       if(diff1>50) {robotright(70);}
       else if (diff1<-50) { robotleft(70);  }
       else
       {
        robotstop(); cnt =2;
        }
       }
    robotstop();
    break;
    }
    }
  }
```

```
}
//int cnt2=0;

void moveleft()
{
 alignfront(350);
 robotstop();delay(500);
 robotslideright(100);delay(400);
 robotstop();
}
int cnt2 = 0;

void loop() {
 //delay(60);
 while(hc.dist()>45.0 )
 {
  moveleft();
 }
 cnt2 = 1;
 robotstop();  delay(500);
 robotleft(100); delay(375);
 robotstop(); delay(500);

 while(hc.dist()>45.0 )
 {
  moveleft();
 }
 cnt2 = 2; robotstop();  delay(500);
 robotleft(100); delay(375);
 robotstop(); delay(500);

  while(hc.dist()>45.0 )
 {
  moveleft();
 }
 cnt2 = 3; robotstop();  delay(500);
 robotleft(100); delay(375);
 robotstop(); delay(500);

  while(hc.dist()>45.0 )
 {
  moveleft();
 }
 cnt2 = 4;
 robotstop();


sini: delay(60);
goto sini;
}
```

b. Arduino Code L-Shape Room

```
#include "Adafruit_VL53L0X.h"
// address we will assign if dual sensor is present
#define LOX1_ADDRESS 0x30
#define LOX2_ADDRESS 0x31
#include <HCSR04.h>

HCSR04 hc(A2, A3);

// set the pins to shutdown
#define SHT_LOX1 A0
#define SHT_LOX2 A1

// objects for the vl53l0x
Adafruit_VL53L0X lox1 = Adafruit_VL53L0X();
Adafruit_VL53L0X lox2 = Adafruit_VL53L0X();

// this holds the measurement
VL53L0X_RangingMeasurementData_t measure1;
VL53L0X_RangingMeasurementData_t measure2;

#include <AFMotor.h>
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

int jarak;
int offset;
int sen1, sen2;

void setID() {
  // all reset
  digitalWrite(SHT_LOX1, LOW);
  digitalWrite(SHT_LOX2, LOW);
//  digitalWrite(SHT_LOX3, LOW);
  delay(10);
  // all unreset
  digitalWrite(SHT_LOX1, HIGH);
  digitalWrite(SHT_LOX2, HIGH);
  //digitalWrite(SHT_LOX3, HIGH);
  delay(10);

  // activating LOX1 and reseting LOX2
  digitalWrite(SHT_LOX1, HIGH);
  digitalWrite(SHT_LOX2, LOW);
  //digitalWrite(SHT_LOX3, LOW);

  // initing LOX1
```

```
  if(!lox1.begin(LOX1_ADDRESS)) {
    Serial.println(F("Failed to boot first VL53L0X"));
    while(1);
  }
  delay(10);

  // activating LOX2
  digitalWrite(SHT_LOX2, HIGH);
  delay(10);

  //initing LOX2
  if(!lox2.begin(LOX2_ADDRESS)) {
    Serial.println(F("Failed to boot second VL53L0X"));
    while(1);
  }

  lox1.configSensor(Adafruit_VL53L0X::VL53L0X_SENSE_LONG_RANGE);
  lox1.configSensor(Adafruit_VL53L0X::VL53L0X_SENSE_LONG_RANGE);

}

void read2sensors() {

  lox1.rangingTest(&measure1, false); // pass in 'true' to get debug data printout!
  lox2.rangingTest(&measure2, false); // pass in 'true' to get debug data printout!

  // print sensor one reading
  Serial.print(F("1: "));
  if(measure1.RangeStatus != 4) {     // if not out of range
   sen1 = measure1.RangeMilliMeter;
    Serial.print(sen1);
  } else {     Serial.print(F("999"));  }
 Serial.print(F(" "));

  // print sensor two reading
  Serial.print(F("2: "));
  if(measure2.RangeStatus != 4) {
    sen2 = measure2.RangeMilliMeter;
    Serial.print(sen2);
  } else {     Serial.print(F("999"));  }
  Serial.println();
}

void robotstop()
{
  motor1.run(FORWARD);    motor1.setSpeed(0);  motor2.run(FORWARD);
motor2.setSpeed(0);
  motor3.run(FORWARD);    motor3.setSpeed(0);  motor4.run(FORWARD);
motor4.setSpeed(0);
}
```

```
void robotfwd(int speed)
{
  motor1.run(FORWARD);    motor1.setSpeed(speed);  motor2.run(FORWARD);
motor2.setSpeed(speed);
  motor3.run(FORWARD);    motor3.setSpeed(speed+40);  motor4.run(FORWARD);
motor4.setSpeed(speed+20);
}

void robotrev(int speed)
{
  motor1.run(BACKWARD);    motor1.setSpeed(speed);  motor2.run(BACKWARD);
motor2.setSpeed(speed);
  motor3.run(BACKWARD);    motor3.setSpeed(speed+40);
motor4.run(BACKWARD);    motor4.setSpeed(speed+20);
}

void robotright(int speed)
{
  motor1.run(FORWARD);    motor1.setSpeed(speed);  motor2.run(BACKWARD);
motor2.setSpeed(speed);
  motor3.run(BACKWARD);    motor3.setSpeed(speed+40);  motor4.run(FORWARD);
motor4.setSpeed(speed+20);
}

void robotleft(int speed)
{
  motor1.run(BACKWARD);    motor1.setSpeed(speed);  motor2.run(FORWARD);
motor2.setSpeed(speed);
  motor3.run(FORWARD);    motor3.setSpeed(speed+40);  motor4.run(BACKWARD);
motor4.setSpeed(speed+20);
}

void robotslideright(int speed)
{
  motor1.run(FORWARD);    motor1.setSpeed(speed);  motor2.run(BACKWARD);
motor2.setSpeed(speed);
  motor3.run(FORWARD);    motor3.setSpeed(speed+40);  motor4.run(BACKWARD);
motor4.setSpeed(speed+20);
}

void robotslideleft(int speed)
{
  motor1.run(BACKWARD);    motor1.setSpeed(speed);  motor2.run(FORWARD);
motor2.setSpeed(speed);
  motor3.run(BACKWARD);    motor3.setSpeed(speed+40);  motor4.run(FORWARD);
motor4.setSpeed(speed+20);
}

void setup() {
```

```
  jarak = 450;
  offset = 50;
  Serial.begin(115200);
  // wait until serial port opens for native USB devices
  while (! Serial) { delay(1); }
  pinMode(SHT_LOX1, OUTPUT);  pinMode(SHT_LOX2, OUTPUT);
//pinMode(SHT_LOX3, OUTPUT);
  Serial.println(F("Shutdown pins inited..."));
  digitalWrite(SHT_LOX1, LOW);  digitalWrite(SHT_LOX2, LOW);
//digitalWrite(SHT_LOX3, LOW);
  Serial.println(F("Both in reset mode...(pins are low)"));
  Serial.println(F("Starting..."));
  setID();

}



void alignfront(int dist)
{
  int cnt=0; int diff1=0; int diff2=0;
  read2sensors();
 while(cnt==0)
  {
    read2sensors(); diff1 = sen2 - sen1;
    if(sen1 > dist && sen2 > dist)    {robotfwd(70);      }
    if (sen1 < dist && sen2 <dist)
    {
      if(sen1 < dist-100 && sen2 < dist-100)    {robotrev(70);      }
    else
      {robotstop(); cnt =1;
      while(cnt == 1)
      {
       read2sensors(); diff1 = sen2 - sen1;
       if(diff1>50) {robotright(70);}
       else if (diff1<-50) { robotleft(70);  }
       else
       {
        robotstop(); cnt =2;
        }
       }
    robotstop();
    break;
    }
    }
  }
}
//int cnt2=0;

void alignfront2(int dist)
{
```

```
 int cnt=0; int diff1=0; int diff2=0;
 read2sensors();
 while(cnt==0)
  {
   read2sensors(); diff1 = sen2 - sen1;
   if(sen1 > dist && sen2 > dist)
   {robotfwd(70); delay(800); robotstop();delay(500);
   robotleft(100);delay(375); robotstop();delay(500);
   robotfwd(70);     }

   if (sen1 < dist && sen2 < dist)
   {
     if(sen1 < dist-100 && sen2 < dist-100)    {robotrev(70);      }
   else
     {robotstop(); cnt =1;
     while(cnt == 1)
     {
      read2sensors(); diff1 = sen2 - sen1;
      if(diff1>50) {robotright(70);}
      else if (diff1<-50) { robotleft(70);  }
      else
      {
       robotstop(); cnt =2;
       }
      }
   robotstop();
   break;
   }
   }
  }
}

void moveleft()
{
 alignfront(350);
 robotstop();delay(500);
 robotslideright(100);delay(400);
 robotstop();
}

void moveleft2()
{
 alignfront2(350);
 robotstop();delay(500);
 robotslideright(100);delay(1000);
 robotstop();
}
int cnt2 = 0;

void loop() {
```

```
 while(hc.dist()> 45.0)
 {
  moveleft2();
 }
 cnt2 = 1;
 robotstop();  delay(500);
 robotright(100); delay(375);
 robotstop(); delay(500);

 while(hc.dist()> 45.0 )
 {
  moveleft();
 }
 cnt2 = 2; robotstop();  delay(500);
 robotleft(100); delay(375);
 robotstop(); delay(500);

  while(hc.dist()> 45.0 )
 {
  moveleft();
 }
 cnt2 = 3; robotstop();  delay(500);
 robotleft(100); delay(375);
 robotstop(); delay(500);

  while(hc.dist()> 45.0 )
 {
  moveleft();
 }
 cnt2 = 4;
 robotstop();


sini: delay(60);
goto sini;

}
```